

國立政治大學資訊管理研究所

指導教授：曾淑峰 博士

以 Scrum 觀念及 SSH 框架為基礎進行系統開發之研究  
A Study on System Development based on Scrum Concept and SSH Framework

研究生：宋洪炘 撰

中華民國一百零一年七月

## 摘要

由於現今商業環境時常變動，使得使用者需求在系統開發過程中頻繁改變。傳統系統開發方式無法迅速回應使用者需求變更，強調彈性開發流程的敏捷開發方式逐漸廣為接受。敏捷開發並未探討系統設計議題而系統設計不佳會導致系統品質低落，使用軟體框架可以快速建立良好的分層結構，有助於穩定產品品質並提高軟體再用性。本研究以 Scrum 為基礎，結合 Struts、Spring、Hibernate 整合框架，建立一套系統開發方法，包含敏捷開發方式的彈性流程，兼顧系統擴充性及可維護性，研究方法為概念形成法與系統展示法，並以 Java Web-Based 離型系統為例，說明相關作業內容。

關鍵詞:Scrum、SSH、軟體架構、敏捷開發、系統開發

## Abstract

Due to the quick variation in the business environment, user requirements change frequently during the system development process. This induces the rising popularity of agile development method which emphasizes the development flexibility during the process. However, agile development method does not address the design issues and poor system design can easily cause poor system quality. Software framework helps to quickly build a good hierarchy structure and thus improve the product quality and software reusability. This study proposes a system development method based on Scrum with the integration of Struts、Spring and Hibernate frameworks. The method is embedded with the development flexibility from the agile method, as well as the system extensibility and maintainability from the software frameworks. The study is conducted with the research method of concept formulation, and would be continued with a Java Web-Based prototype development as an example to illustrate the relevant activities.

Keyword: Scrum、SSH、Software Architecture, Agile Development, System Development

# 目錄

第一章 緒論.....	1
第一節 研究動機.....	1
第二節 研究目的.....	2
第三節 研究方法.....	3
第四節 研究範圍與限制.....	4
第五節 研究架構.....	5
第六節 論文架構.....	6
第二章 文獻探討.....	7
第一節 軟體工程.....	7
第二節 開發框架.....	21
第三節 UML.....	29
第四節 人力資源.....	32
第三章 系統發展研究架構.....	34
第一節 初始階段.....	35
第二節 系統高階規劃階段.....	36
第三節 反覆執行階段.....	40
第四章 系統發展實作-以人力資源管理系統雛型為例.....	44
第一節 初始階段.....	44
第二節 系統高階規劃階段.....	49
第三節 反覆執行階段.....	72
第五章 結論與未來發展方向.....	81
參考文獻.....	83

# 圖目錄

圖 1-1 研究架構.....	5
圖 1-2 論文架構.....	6
圖 2-1 瀑布模式.....	7
圖 2-2 螺旋形模式.....	8
圖 2-3 RUP 模式.....	9
圖 2-4 SCRUM 流程圖.....	13
圖 2-5 SCRUM 結合各種最佳實務.....	15
圖 2-6 STRUTS 框架.....	21
圖 2-7 SPRING 框架.....	24
圖 2-8 HIBERNATE 框架.....	26
圖 2-9 SSH 整合框架結構圖.....	27
圖 2-10 SSH 整合框架流程圖.....	28
圖 3-1 系統開發程序及方法.....	34
圖 3-2 JAVA 普及框架比例圖.....	36
圖 3-3 類別範例圖.....	39
圖 4-1 系統使用者案例圖.....	46
圖 4-2 系統故事.....	47
圖 4-3 系統產品待辦清單.....	48
圖 4-4 系統主流程活動圖.....	49
圖 4-5 STRUTS2 流程圖.....	51
圖 4-6 STRUTS2 過濾器.....	52
圖 4-7 STRUTS2 過濾器編碼.....	52
圖 4-8 STRUTS2 STRUTSPREPAREANDEXECUTEFILTER.....	53
圖 4-9 STRUTS2 攔截器.....	54

圖 4-10 STRUTS2 攔截器串.....	54
圖 4-11 JSTL 使用方式.....	55
圖 4-12 BEAN 作用域配置.....	56
圖 4-13 原 SPRING XML 配置.....	57
圖 4-14 新 SPRING XML 配置.....	57
圖 4-15 HIBERNATE 配置文件.....	58
圖 4-16 HIBERNATE 映射文件.....	58
圖 4-17 JAVASCRIPT 框架使用比例.....	59
圖 4-18 JQUERY 選擇器使用方式.....	60
圖 4-19 SPRING 與 STRUTS 整合.....	60
圖 4-20 SPRING 實例化 STRUTS ACTION.....	60
圖 4-21 SPRING 管理 HIBERNATE 資料庫配置.....	61
圖 4-22 SPRING 管理 HIBERNATE SESSIONFACTORY.....	61
圖 4-23 SPRING 管理 HIBERNATE TRANSACTIONFACTORY.....	62
圖 4-24 JQUERY 與 SSH 整合方式.....	62
圖 4-25 系統資料庫 ERD 圖.....	63
圖 4-26 系統登入功能類別圖.....	65
圖 4-27 系統基礎設施.....	66
圖 4-28 系統檔案結構圖.....	67
圖 4-29 系統分層循序圖.....	68
圖 4-30 團隊分工.....	72
圖 4-31 衝刺目標與長度.....	72
圖 4-32 MAVEN 目錄結構.....	74
圖 4-33 衝刺待辦清單.....	75
圖 4-34 任務板.....	76

圖 4-35 系統登入功能 .....	77
圖 4-36 員工資料新增功能 .....	77
圖 4-37 員工資料修改功能 .....	78
圖 4-38 員工資料刪除功能 .....	78
圖 4-39 員工資料動態條件查詢功能 .....	79
圖 4-40 系統登出功能 .....	79
圖 4-41 精製產品待辦清單 .....	80
圖 4-42 衝刺回顧 .....	80



# 表目錄

表 2-1 SCRUM 角色與負責工作.....	12
表 2-2 SCRUM 各項會議.....	12
表 2-3 SCRUM 產出物 .....	13
表 2-4 最佳實務的定義及範例.....	15
表 2-5 軟體開發模型比較.....	19
表 2-6 STRUTS 提供 MODEL VIEW CONTROLLER 的相關實作 .....	22
表 2-7 UML 圖形分類.....	30
表 2-8 人力資源管理理論 .....	32
表 2-9 人力資源資訊系統 .....	33
表 3-1 故事描述表.....	37
表 3-2 PACKAGE 範例分類表 .....	37
表 4-1 故事描述表.....	50
表 4-2 BEAN 作用域分類表 .....	56
表 4-3 SPRING2.0 註釋標籤.....	57
表 4-4 資料庫欄位說明表 .....	64
表 4-5 系統 PACKAGE 分類表.....	66
表 4-6 系統 SOURCE 目錄結構 .....	69
表 4-7 系統開發環境.....	69
表 4-8 系統共用函式庫 .....	70



## 誌謝

回想當年剛進碩士班的懵懵懂懂，到現在的碩士論文完成，在碩士班的期間認識了很多研究所的好朋友們，像是葉亦宸、陳颯潔、賴秉宏以及學弟林佑儒等等...可以和大家一起學習討論是一件十分令人開心的一件事。

在專業方面以及這篇論文能順利完成，則必須感謝指導教授曾淑峰的熱心指導，使得這篇論文可投上 ICIS 研討會，以及最後的論文提報。另外也要感謝論文的口試委員陳宗天教授以及季延平教授的不辭辛勞指導，也給予此篇論文許多專業上改進之意見。

最後能讓我無後顧之憂的完成此篇論文以及順利念完研究所，則必須感謝提供我生活費以及支持我唸研究所的家人，也要謝謝黃怡婷教導我如何編排文件，在文思枯竭時陪我聊天以及給我支持，其實要感謝的人與事很多，幾句話語無法完全表達我內心的感謝，真的謝謝你們。

# 第一章 緒論

## 第一節 研究動機

在軟體工程的領域，從過去的瀑布式模型、漸增式模型至螺旋模型，皆是以預先規劃好流程方式來執行整個專案，往往無法立即的反應臨時性的需求變更，冗餘的文件產出也對開發團隊帶來莫大的無形壓力。近年來，軟體工程的重心開始轉變，更加強調產品的產出應該是要符合顧客的期望，所以開發團隊應該追求改善產品品質以及提升交付產品的速度，並且考量專案開發期間的成本效益是否符合預期目標，但在改善流程中常常造成整個專案的複雜度增加，開發成本不降反升。為了面對這困難的議題，以人與人之間彈性互動為主軸的敏捷開發方法漸被重視，依據由 Stabdish Group 的調查結果顯示，敏捷專案每年可使專案效能成長 22%，但傳統的瀑布式專案每年僅能使專案效能成長 1%(Williams et al. 2011)。

敏捷開發方式除了強調團隊中人與人之間的互動，不良設計所造成程式碼的缺失也是該關注的一個議題，由 Martin(2003)所提到像是僵化性(Rigidity)、脆弱性(Fragility)、固定性(Immobility)、黏滯性(Viscosity)、不必要的複雜度(Needless Complexity)、不必要的重複(Needless Repetition)以及晦澀性(Opacity)等不良設計症狀，皆會導致專案窒礙難行，嚴重甚至會導致專案面臨失敗，需要透過一些方法來盡量消弭這些壞味道。許多軟體開發者以及研究員共同提出了一套物件導向設計的原則，來減少壞味道的發生，這些原則分別是單一責任原則、開放封閉原則、Liskov 替代原則、相依性反向原則以及介面分割原則，但要實際設計出符合這些原則的最佳軟體結構，是一件進入門檻極高且需要投入大量的時間來進行設計工作，對於有交付時間壓力的專案是一個沉重的負擔，故我們需要一個可以快速建立起優良軟體架構的工具或方法來提升專案產品的品質以及專案的成功率。

## 第二節 研究目的

在使用哪種敏捷方式的調查中,2008年6月和7月間80個不同國家3061個受訪者,有49%是使用Scrum(Williams et al. 2011),這顯示Scrum為近年來熱門的一種敏捷專案開發方法。為了保持Scrum本身的彈性使用,在技術方面並未有詳細的規範,可能造成軟體架構的壞味道產生。框架,是一組抽象類別、實體類別和相關服務的集合,它是一個應用系統的通用骨架,透過框架我們可以快速的建立起軟體分層以及符合物件導向設計原則的軟體架構(葉文崎 2002)。Struts、Spring、Hibernate 框架為 J2EE(已更名為 JavaEE,但一般稱呼仍常延用 J2EE)中的主流框架,由於每個框架各有其優缺點,整合三個框架可以互相補足各框架不足之處並可使系統開發更具效率(Hao and Tang 2010)。本研究將以 Scrum 敏捷開發方式為核心,並透過 SSH(Spring, Struts, Hibernate)整合框架來減少程式碼的重複撰寫,降低系統耦合力並提昇系統內聚力,使系統的後續擴展性增加與維護性的成本降低,再透過實作指引文件(Implementation Guide)來解決框架進入障礙,團隊成員在反覆開發迴圈中做細部的設計,並以人力資源管理離型系統為例,說明相關作業內容。作為未來開發類似專案時的重要參考。

### 第三節 研究方法

資管領域之研究方法，國內外學者皆有提出眾多不同看法，像是 Scott Morton(1984)將研究方法分成九大類型：

1. 雛形(Prototype)：建立一個原型系統，模擬真實世界系統運作的概況，可以在此原型逐步修正或增強其功能。通常此種方式可以用在理論的可行性研究。
2. 方法論(Methodology)：當所進行研究的領域，在程序上由嘗試錯誤法(Trial and Error)來歸納有困難，或在理論上無法直接求得時，研究者可以建立一個方法論來進行研究概念的陳述。如 Sprague & Carlson 提出在決策支援系統的建構方法論 ROMC(Representations, Operations, Memory Aids, and Control Mechanism)即為一代表。
3. 理論發展(Theory Development)：要發展一個全新的理論不是一蹴而就的，研究者需要對所研究的課題非常有創見，並且對現行已有的理論要有非常深入的了解及心得。
4. 概念形成(Concept Formulation)：在此一方法中，研究者必須提出一套完整的組織概念(Organization Ideas)，即建議行動(Actions)，以建立一個有用的架構(Framework)。
5. 實驗室實證測試(Empirical Laboratory Test)：在實驗室中以測試的方式模擬真實世界的行為模式。
6. 真實世界實證測試(Empirical Real-World Test)：此一方法是以在真實世界中實證的方式來評估新觀念，新方法論或新系統的效益。
7. 調查(Survey)：透過對某一特定統計目標之母體(Population)進行調查，以統計數據的方式來了解某一事件的存在性。
8. 個案(Case)：在研究之中完整地描述一個個案，以對此一應用領域提供一個深度了解及掌握實際狀況的環境。
9. 真理(Truth)：即使在缺乏強而有力的理論及實證做為後盾的情況下，研究者

仍可提出所謂的真知卓見，使人一見即心領神會，奉為圭臬。

國內學者梁定澎(1997)則指出研究方法可分為研究(Case Study)、調查研究(Survey Research Method)、實驗性研究(Experimental research method)、模式推導(Modeling)系統展示(System Representation)、彙總研究(Meta-Research)六種。

由於本研究以軟體工程中的敏捷系統開發方式做為基礎觀念並搭配開發框架進行資訊系統雛形的開發，所以本研究將採取 Scott Morton 所提出之概念形成(Concept Formulation)以及梁定澎所提出之系統展示(System Representation)兩種方式進行研究。

#### **第四節 研究範圍與限制**

本研究採用概念形成法與系統展示法進行研究，對於敏捷開發方法與框架技術之採用源自於文獻探討與實務觀察，並未直接進行調查分析，對於研究貢獻還無法驗證。有待提出這樣的組合概念與雛型說明之後，持續擴充，再從使用實務體驗中進行調查研究。

## 第五節 研究架構

本論文之研究架構如下：

- 一、提出研究動機
- 二、對軟體工程進行文獻探討
- 三、對所應用到之開發框架相關技術進行文獻探討
- 四、對 UML 進行文獻探討
- 五、對人力資源進行文獻探討
- 六、提出一個系統開發模式
- 七、依照所提出的模式開發系統雛型架構
- 八、結論與未來發展方向

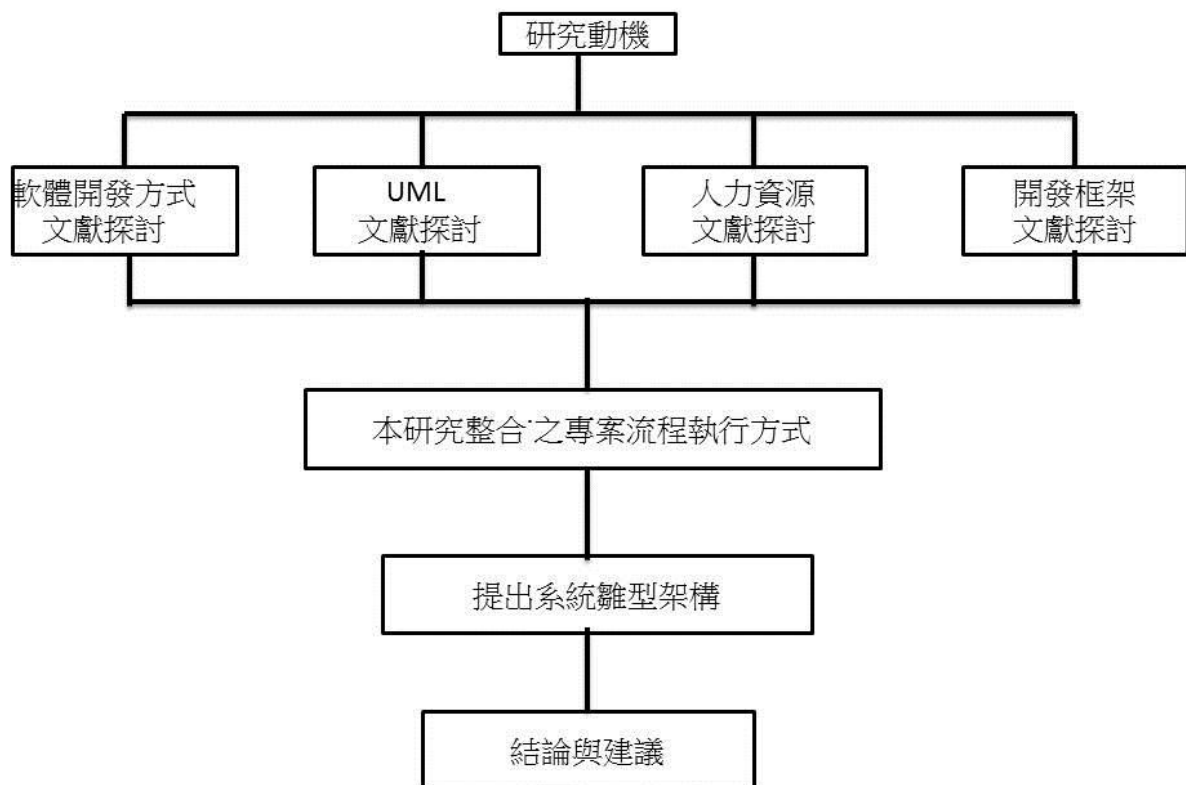


圖 1-1 研究架構

## 第六節 論文架構

本論文章節如下圖：

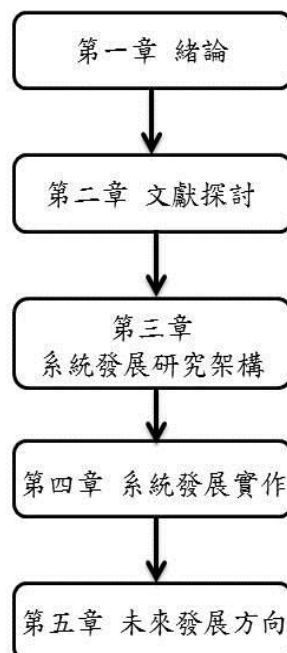


圖 1-2 論文架構

## 第二章 文獻探討

### 第一節 軟體工程

#### 一、傳統軟體開發方式

##### 1. 瀑布模式

由 W.W.Royce 在 1970 年提出 Waterfall 模型，他認為開發需按照需求分析，設計，實作，測試，上線及維護純線性進行。

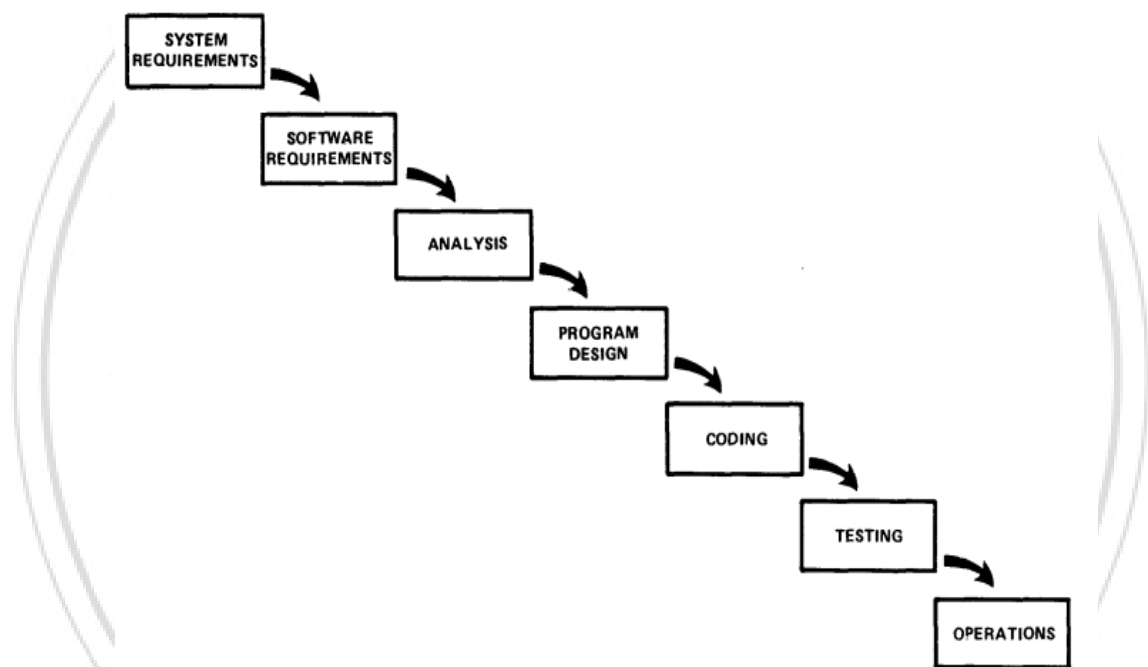


圖 2-1 瀑布模式

資料來源：Royce, W. W. "Managing the development of large software systems", Proc. Westcon, IEEE CS Press, 1970, pp. 328-339.

瀑布模型（Waterfall Model）又稱為系統發展生命週期，強調系統開發應有定義清楚的階段，且必須完整的經歷週期之每一開發階段，在每一階段結束後產生大量文件，為一種線性化的系統開發方式。



## 2. 漸增模式

漸增模式(Mills 1971)為瀑布模式的擴充，漸增模式是將系統劃分成數個子系統或功能，每個子系統或功能可獨立開發，直到系統完整為止。在開發過程中每次只交付某個子系統或功能。

## 3. 螺旋形模式

美國軟體工程師 Barry Boehm 於 1988 年提出螺旋模型。

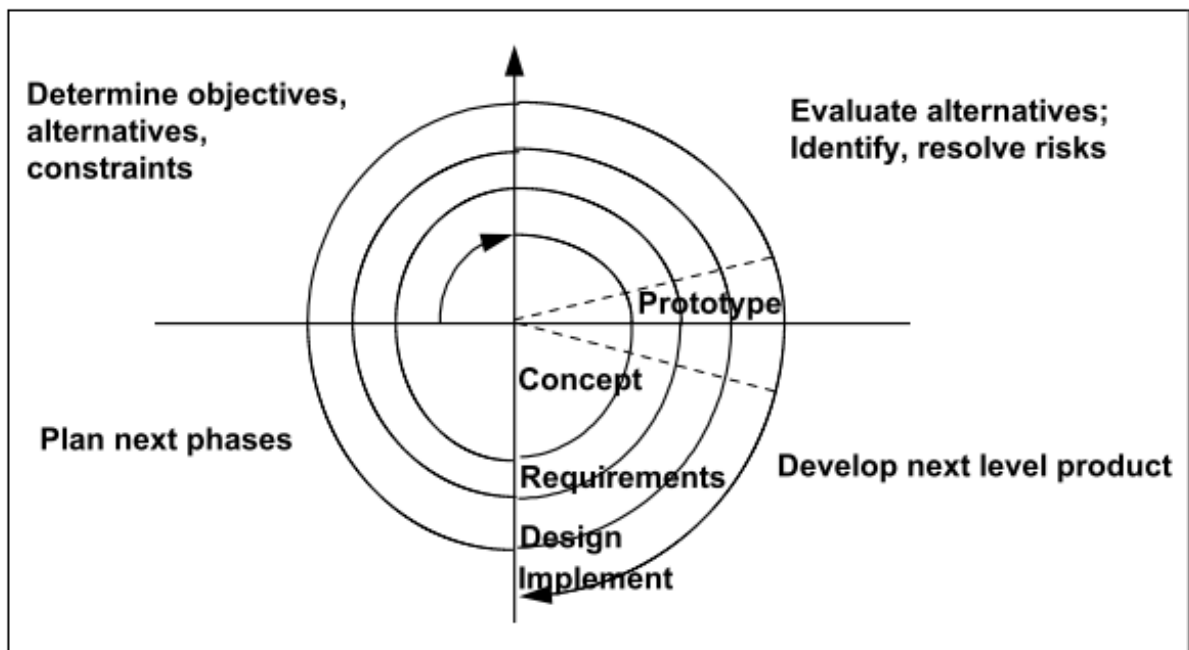


圖 2-2 螺旋形模式

資料來源：Boehm, B. W. "A Spiral Model of Software Development and Enhancement," IEEE Computer (21:5), 1988, pp. 61-72.

螺旋模型加入了其他模型沒有的風險分析，由以下的步驟構成：

- (1) 確定軟體的目標，決定實施方案以及備選方案並弄清楚開發限制條件。
- (2) 分析評估所選方案，辨別所存在的風險以及思考如何降低消除風險。
- (3) 在可接受的風險範圍內，進行軟體開發。
- (4) 由客戶評價此階段開發工作，並對下一階段進行計劃與部署。

#### 4. 統一開發過程(俞君翰 2002)

1999 年 RUP 集合了各種方法論，由 UML 的三位創始人 Grady Booch 、 James Rumbaugh 及 Ivar Jacobson 共同合作提出，它所定義的是：「在軟體專案進行當中，應有哪些參與人員，而他們誰應在何時、如何做哪些工作，以共同完成軟體系統的開發。」

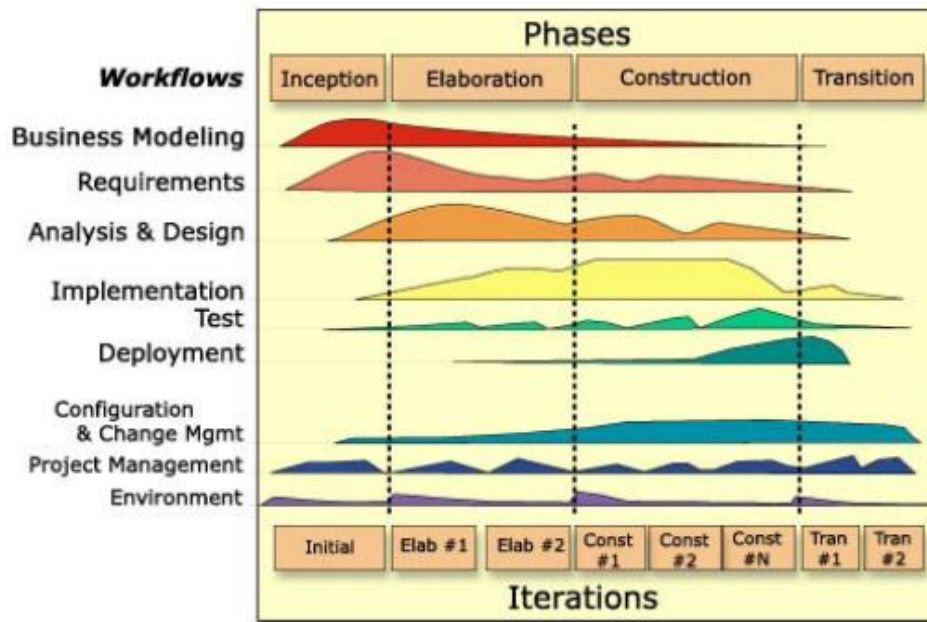


圖 2-3 RUP 模式

資料來源：俞君翰，2002，應用設計樣式於元件式系統開發之研究--以會計總帳系統為例，國立政治大學資訊管理研究所碩士論文。

RUP 主要開發流程可分為四階段：

##### (1) 初始階段

此階段目標是替所要開發系統進行可行性分析，定義專案大小及涵蓋範圍，評估專案所需的時間、經費及所需投入的人力。

##### (2) 精製階段

精製階段的目標是分析問題領域，擬定專案計畫，確認商業模型，透過系統分析與設計獲知所需要的功能及非功能性需求，建立健全的系統架構，準備

所需開發的各項工具，並淘汰專案中最高風險的元素。

### (3) 建構階段

建構階段是對所開發的產品進行單元以及整合等測試，並產出 beta 版本的產品。此過程中最重要的是管理資源以及成本、進度和質量的最佳化，並確定軟體、環境、用戶是否可以開始系統的運作。

### (4) 交付階段

為了確保最終產品對用戶是可用的，故將產品分批交給客戶進行測試，並對於客戶的反饋進行少量調整，此時要決定目標是否已達到，若未達到是否開啟另一個開發週期，若達到則對終端客戶進行使用者訓練。

過去的開發方式，像是最傳統具有規劃、分析、設計、建置、操作、維護、控制以及稽核等線性化流程的瀑布模式(Waterfall Model)(Royce 1970)，之後的漸增模式(Incremental Model)、螺旋模式(Spiral Model)(Boehm1988)、Rational 統一開發過程(Rational Unified Process, RUP)(Kruchten2004)，都具有固定的流程以及大量的文件產出，在面對現今多變的商業環境，經常性需求變更的挑戰越來越高，加上使用者與開發者的電腦化經驗已越來越普遍，使用需求的溝通模式越來越直覺，因此彈性的開發流程逐漸普及。

## 二、敏捷開發方式

2001 年，有一群業界的專家，聚集在一起，以自身的經驗以及事例，提出了一些讓軟體開發團隊可以迅速工作以及回應變化的價值觀和原則，稱為「敏捷軟體開發宣言」(The Manifesto of the Agile Alliance)，並提出了敏捷軟體開發之 12 原則(Beck et al. 2001)。

在「敏捷軟體開發宣言」中，包含了四項宣言，分別為個人及互動勝於流程與工具、可用的軟體勝於詳盡的文件、與客戶合作勝於合約談判以及回應變更勝於墨守計畫。而敏捷軟體開發 12 個原則如下所述：

- 最優先的任務，是透過及早並持續地交付有價值的軟體來滿足客戶需求。
- 竭誠歡迎改變需求，甚至已處開發後期亦然。敏捷流程掌控變更，以維護客戶的競爭優勢。
- 經常交付可用的軟體，頻率可以從數週到數個月，以較短時間間隔為佳。
- 業務人員與開發者必須在專案全程中天天一起工作。
- 以積極的個人來建構專案，給予他們所需的環境與支援，並信任他們可以完成工作。
- 面對面的溝通是傳遞資訊給開發團隊及團隊成員之間效率最高且效果最佳的方法。
- 可用的軟體是最主要的進度量測方法。
- 敏捷程序提倡可持續的開發。贊助者、開發者及使用者應當能不斷地維持穩定的步調。
- 持續追求優越的技術與優良的設計，以強化敏捷性。
- 精簡或最大化未完成工作量之技藝是不可或缺的。
- 最佳的架構、需求與設計皆來自於能自我組織的團隊。
- 團隊定期自省如何更有效率，並據之適當地調整與修正自己的行為。

## 1. Scrum 基本流程與產出

Scrum 為敏捷開發方式的其中一種。Schwaber (2007)指出 Scrum 為一種反覆漸增式軟體開發過程，和傳統增量式開發不同的地方是由一個稱為產品負責人的角色擔任客戶需求的了解，產品負責人在整個團隊開發期間都會參與，客戶也會在每一次的衝刺檢視會議中檢視每一次所增量的項目是否符合心中期待，此外 Scrum 更強調人與人之間的互動，透過不同型式的會議來提升團隊內的信任以及士氣，也因為這樣提升了知識的傳播性因而減少大量正規文件的製作。由下表介紹 Scrum 中的角色與所負責工作：

表 2-1 Scrum 角色與負責工作

角色	工作
產品負責人 (Product Owner)	決定產品的功能性需求，並定義產品的交付時間、完成的定義。
Scrum 教練 (Scrum Master)	消除團隊執行 Scrum 時的障礙，負責團隊成員間的協調與溝通，並設法改善 Scrum 的流程。
開發團隊成員 (Team Members)	人數約 5~9 人，團隊成員熟悉產品開發所需的各項技術。

資料來源:本研究整理

下表顯示 Scrum 過程中的各項會議：

表 2-2 Scrum 各項會議

會議名稱	會議內容
發布規劃會議 (Release Planning Meeting)	由產品負責人和客戶討論系統所需要的功能性需求，並依需求找尋所需的開發團隊成員，組成開發團隊小組。
衝刺規劃會議 (Sprint Planning Meeting)	決定下一個衝刺期間所需完成的故事。
每日 Scrum 會議 (Daily Scrum Meeting)	每天利用 15 分鐘來進行，由 Scrum Master 詢問每個開發成員三項問題，分別是： (1) 你昨天做了些什麼？ (2) 你今天打算做些什麼？ (3) 昨天進行的過程中是否有遇到任何阻礙或問題？
衝刺檢視會議 (Sprint Review Meeting)	用以展示所完成的工作項目。
衝刺回顧會議 (Sprint Retrospective Meeting)	檢討此次的衝刺該如何做改善，並對下一次的衝刺做適當的調整，另外此次尚未做完的故事或任務都將加回產品待辦清單中，由下一次衝刺執行。

資料來源：Sutherland, J., Schwaber, K. “The Scrum Papers: Nuts, Bolts, and Origins of an Agile Process.,” Draft, 2007 (available online at <http://scrumtraininginstitute.com/>).

下表顯示 Scrum 過程中的各項產出：

表 2-3 Scrum 產出物

產出物名稱	產出物內容
產品待辦清單 (Product Backlog)	包含客戶所需要的功能性需求，非功能性需求則以技術故事的方式加入產品待辦清單中。
衝刺待辦清單 (Sprint Backlog)	在每次衝刺規劃會議所決定要執行的項目紀錄。
燃盡圖 (Burndown Chart)	呈現一個時間區段中，所剩餘的工作量，使 Scrum Master 以及產品負責人可對專案工作進行有所掌握。

資料來源：Sutherland, J., Schwaber, K. “The Scrum Papers: Nuts, Bolts, and Origins of an Agile Process.,” Draft, 2007 (available online at <http://scrumtraininginstitute.com/>).

Scrum 流程如下圖：

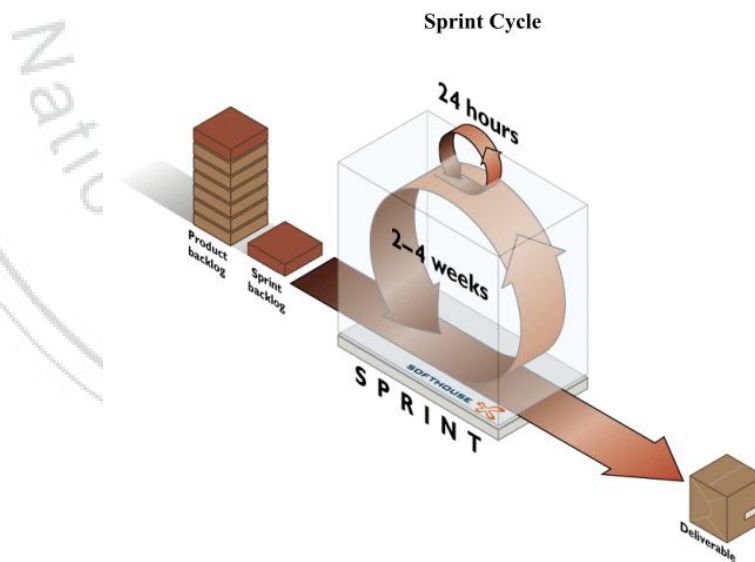


圖 2-4 Scrum 流程圖

資料來源：Sutherland, J., Schwaber, K. “The Scrum Papers: Nuts, Bolts, and Origins of an Agile Process.,” Draft, 2007 (available online at <http://scrumtraininginstitute.com/>).

產品負責人和客戶討論後，將初步功能性需求寫在產品待辦清單中。再和開發

團隊開衝刺會議，決定衝刺週期的長度，以及從產品清單中的多項產品項目中，依重要性來決定這一次衝刺所要執行的故事數，並記錄於衝刺待辦清單。

衝刺執行時期，每天會舉行 15 分鐘會議，由 Scrum Master 主持，詢問團隊成員 3 個問題：(1)你昨天做了什麼？(2)你今天打算做什麼？(3)你在昨天執行的過程中有無遇到任何阻礙或問題？。在衝刺期間，原先訂定的衝刺目標是不可更改的，若有需要則選擇取消此次衝刺或是在衝刺回顧會議中作產品待辦清單的修正。

衝刺執行結束後，會產生一個增量的交付產品，此時邀請客戶以及此專案的相關利害關係人來一起檢視此階段結果展示，再由開發團隊自行開一個檢討會議，來檢討此次衝刺的缺失，並可對下一次的衝刺有適當的修正。

## 2. Scrum 結合最佳實務

Scrum 為一個開放彈性架構，它並未對所有流程做詳細的規定，可讓開發團隊有足夠的彈性去面對不同的情形，以創造出最適合自己團隊的 Scrum 開發流程 (Rising and Janoff 2000)。

Scrum 方面可以依自己團隊情形做調整的項目如下：

- (1) 衝刺長度：可依據不同情況，制定最適當的衝刺時間長度，但以不超過 4 周為主。
- (2) 每日 Scrum 次數：有些團隊無法每天執行每日 Scrum 會議，將視情況修改一個禮拜開會次數，像是兩天開一次會，但盡量不要一個禮拜少於 3 次每日 Scrum 會議。
- (3) 完成定義：每項功能交付都應該事先定義如何為完成，這項工作由產品負責人完成。

Scrum 在結合最佳實務的情形下，可以提升管理效率以及產品品質 (Williams et al. 2011)，但結合最佳實務並不是一次全部就加入所有最佳實務，應該是以循序的方式，逐漸讓開發團隊熟悉再逐步導入 (張嘉琪，2010)。

下圖為 Scrum 結合各種最佳實務：

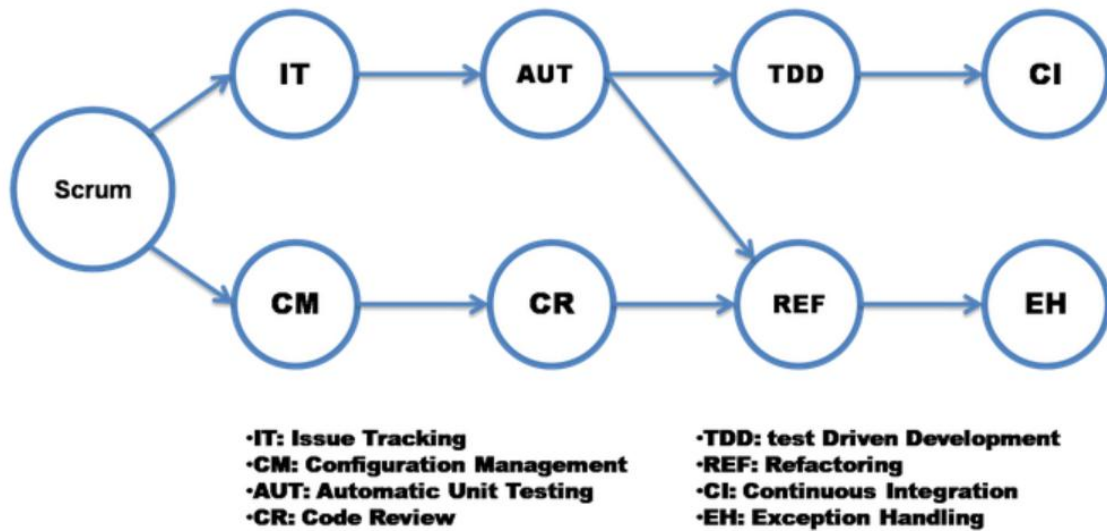


圖 2-5 Scrum 結合各種最佳實務

資料來源：ezScrum，2011，<http://scrum.tw/index.php/tw/home>

下表為本研究透過 ezScrum 網站，以及研究文獻 Scrum+ Engineering Practices: Experiences of Three Microsoft Teams (Williams et al. 2011)、一個以 Scrum 為基礎的軟體工程實務導入方法(張嘉琪，2010)以及 JCIS:支援平台相依性建構之 Java 持續整合系統(吳家豪 et al. 2007)所彙整出的最佳實務定義以及範例表格：

表 2-4 最佳實務的定義及範例

名稱	定義(資料來源)	實際應用例子	工具
建構管理 (Configuration Management)	軟體產出物與釋出的版本規劃與管理作法。列管的對象有軟體功能所依據的相關技術規章、準則、法令；軟體專案的產出物如需求、分析、設計、程式碼、測試文件、部署	將文件、程式等資訊儲存到一個 Source Control System 中，使每個開發者都可以檢視程式碼。只有管理者可以整合到最主要的枝幹(main branch)。存取權限也必須控管，使得某	SVN、CVS、Collabnet Subversion Edge



	文件、維護手冊、使用說明等，專案團隊所使用的開發工具、測試與運作環境的規格與建立辦法等。	些人可以讀寫修改程式碼，另外一些人只可以檢閱程式碼。	
議題追蹤 (Issue tracking)	軟體生命週期中，軟體團隊所面臨的議題、客戶回報的錯誤，或是足以影響進度、成本、品質的事件等的列管、處理、與管控。	將每日的議題由 Scrum Master 輸入進議題追蹤系統，此方式也可以 Skype 工具取代，減少輸入進議題追蹤系統的時間浪費。	Mantis、 BugTracker
自動化 單元測試 (Automatic unit testing)	自動化單元測試的目的是對每一個工作單元 (unit)，如一個副程式 (method)、物件的方法 (operation) 等進行自動化的測試，瞭解其運行結果是否符合預期。	利用靜態分析工具在開發過程初期辨識出共通的程式問題。但如果不是執行初期則會造成開發人員額外負擔，因為會有許多 warning 產生。 另外開發人員也被要求管理自己的自動化測試涵蓋程度，若定義過高的涵蓋程度，也會造成開發人員要強迫測試程式去跑錯誤的條件，影響整體開發效率。	JUnit、 CPPUnit
程式碼檢閱 (Code review)	一個檢測程式碼的程序，用來檢驗程式碼中是	透過資深的開發者來檢閱程式碼和設計架構。	Jupiter、JCR

	<p>否有隱藏的問題，或是可能出錯的地方，例如記憶體洩漏(memory leak)、緩衝區滿溢(buffer overflow)等；同時也可以檢閱有沒有設計的瑕疵，經由這樣的檢閱程序，可以提升軟體的品質與開發人員的程式設計技巧。</p>		
<p>測試驅動開發 (Test-Driven Development)</p>	<p>在開發過程中，程式設計師必須先撰寫單元測試，利用單元測試定義應用程式的介面設計，再開發合乎需求的應用程式。</p>	<p>開發使用這種方式可以大量減少產品缺陷，讓處理 bug 的時間可以大幅的轉移到開發的商業邏輯上面，但這方法的學習進入門檻較高。</p>	<p>JUnit</p>
<p>重構 (Refactoring)</p>	<p>以不影響功能的方式改變程式的原有架構或設計的作法，目的為提升程式碼(code)的品質。</p>	<p>利用設計樣式(Design Pattern)或框架方式來為原有的程式做改造，減少重複的程式碼，提升程式架構的彈性，一般由較有經驗的程式設計師負責。</p>	
<p>持續整合 (Continuous Integration)</p>	<p>一種透過自動化的方式，持續整合軟體系統中各個模組與元件，以保持系統穩定度與品質的軟體</p>	<p>在專案執行的地方架設一台持續整合伺服器，讓伺服器不斷去向版本管理伺服器索取最新版本並</p>	<p>CruiseControl、 Maven、 JavaCIS</p>

	開發最佳實務作法。	建置編譯出來，由測試人員定時做測試，以保證最新可交付版本的可執行性。	
例外處理 (Exception handling)	程式執行中遭遇預期或非預期問題時，程式的因應方式；作法上可分為告知使用者、記錄、自動重試(retry)或結合系統層次設計提供容錯的等作法。	規劃系統對例外處理的設計，依照複雜度切分成多個故事，加入產品待辦清單中，每回合衝刺執行前，挑選數個例外處理有關的故事加入此回合衝刺。	RLMET、EpiDefactor

資料來源：本研究彙整

### 三、軟體開發模型比較

下表顯示不同軟體開發方式優缺點比較:

表 2-5 軟體開發模型比較

軟體開發模型	優點	缺點
瀑布模式 (Waterfall Model)	<ol style="list-style-type: none"> <li>1. 標準的發展程序，有完整的規劃、分析、設計、測試以及文件管理，可有效控制系統品質。</li> <li>2. 易於分工。</li> <li>3. 符合模組化概念。</li> </ol>	<ol style="list-style-type: none"> <li>1. 線性方式開發，造成若有某階段工作延誤，後續所有工作也因此無法進行。</li> <li>2. 系統未完成前無法看到成果，提高開發風險。</li> </ol>
漸增模式 (Incremental Model)	<ol style="list-style-type: none"> <li>1. 可靈活面對需求的變化。</li> <li>2. 減輕瀑布式開發的風險。</li> </ol>	<ol style="list-style-type: none"> <li>1. 大量的設計文件。</li> <li>2. 軟體開發過程中，客戶僅參與需求分析。</li> <li>3. 子系統切割問題。</li> </ol>
螺旋模式 (Spiral Model)	<ol style="list-style-type: none"> <li>1. 利用雛型確定開發時的明確方向。</li> <li>2. 每一迭代皆有執行風險分析。</li> <li>3. 保留瀑布模型循序前進的優點。</li> </ol>	<ol style="list-style-type: none"> <li>1. 風險分析需要專業知識，倘若風險分析出現偏差則會造成重大損失。</li> <li>2. 進化過程不易控制。</li> </ol>
統一流程 (RUP Model)	<ol style="list-style-type: none"> <li>1. 結合眾多方法論，提供完整軟體開發流程。</li> <li>2. 系統分析與設計使用 UML 來溝通，降低開發團隊與客戶間的溝通隔閡。</li> <li>3. 對於系統開發各階段都有定義及分配每個角色的任務及責</li> </ol>	<ol style="list-style-type: none"> <li>1. 由於綜合眾方法論，導致整個流程過於龐大，學習曲線攀升。</li> <li>2. 支援工具受限於 Rational 自己的產品。</li> <li>3. 耗費大量時間。</li> </ol>

	任。	
Scrum 模式 (Scrum Model)	<ol style="list-style-type: none"> <li>1. 專案進度非常透明化。</li> <li>2. 定期的反省會議，不斷改良做法。</li> <li>3. 對於需求的改變，可快速回應。</li> <li>4. 增量的開發方式，可使客戶安心，並且確定開發方向。</li> </ol>	<ol style="list-style-type: none"> <li>1. 只提供管理的建議，缺少開發技術上的意見。</li> <li>2. 關於 backlog 如何製作管理的文件過少。</li> <li>3. 團隊成員需具備應付專案所有面向的技術能力。</li> </ol>

資料來源：本研究整理



## 第二節 開發框架

### 一、MVC

MVC 屬於一種設計樣式，它將互動的系統分割成三個元件，每一個元件都有其特別的任務。Model 包含了應用資料和管理商業邏輯的功能。View 管理的是 Model 的顯示以及 Model 處理完使用者的需求所傳回的回饋顯示。Controller 負責攔截者用者的輸入，並指定對應的 Model 做處理，最後將 Model 傳回之回饋指定適當的 View 做顯示 (Liu et al. 2008)。

### 二、Struts 框架(Feng and Le 2009)

Apache Struts Framework 是一個開源(open-source)的框架，它的作用為以 MVC 設計樣式為基礎建立 Java Web-Based 應用程式。它的核心是一個由標準技術像是 Java Servlet, Java Beans, Resource Bundles 和 xml 以及各種不同的 Jakarta Common Packages 所建立的彈性控制層。由於 Struts 框架增強 MVC 的相容性且包含豐富的標籤庫，使得近年來許多大型軟體系統都採用 Struts 框架。

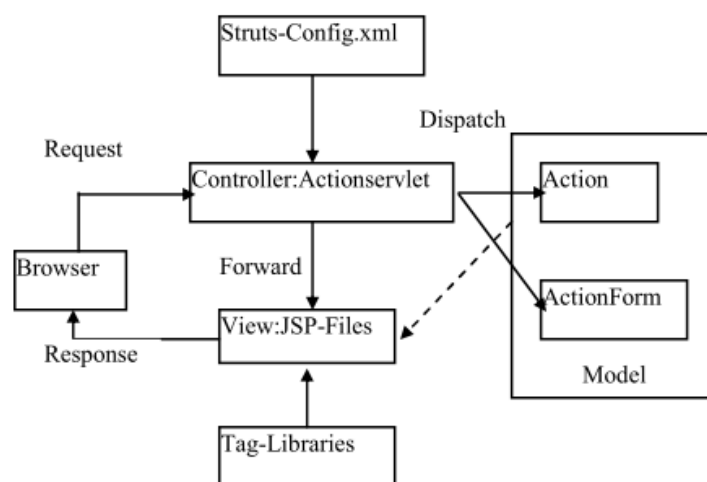


圖 2-6 Struts 框架

資料來源: Feng, X., and Le, T. "Construction of B2B Electronic Commerce System Based on Apache Struts Framework," International Conference on Services Science, Management and Engineering, 2009, pp

221-224.

Struts 的流程為使用者藉著瀏覽器送出請求，由 ActionServlet 獲取請求，ActionServlet 將會去找他相對應的配置檔(Struts-config.xml)，RequestProcessor 則從 ActionMapping 中獲知所要使用來存取資料的 ActionForm 以及回應相對動作的 Action，在處理完後將回應給使用者。

表 2-6 Struts 提供 Model View Controller 的相關實作

Model	負責應用程式的商業邏輯(business logic)。它封裝了資料存取和提供可重複利用的類別函式庫(class library)。Struts 提供 Action 和 ActionForm 給 Model。所有 Action 的物件都是由 Action 衍生而來的子類別。Action 物件將商業邏輯包裝並且回應給 View。ActionForm 能透過定義的屬性來描述使用者資料。
View	控制觀看的資料並且提供使用者收集資料的能力。在 Struts 是用 jsp 來實作 View。Struts 另外提供了多元的標籤函式庫(tags library)，使得它可以輕易的和 Model 做互動。
Controller	用來做 View 和 Model 之間的接合。它收集來自 View 的輸入資料和決定執行的應用程式。Controller 呼叫 Model 並攔截回復的資料，再傳給 View 做更新。在 Struts 中，Controller 由 ActionServlet 和 ActionMapping 組成。ActionServlet 是 Controller 的核心，它包含了一組 ActionMapping，每個 ActionMapping 做 request 和 model 的 Action 互相對應的動作。

資料來源: Feng, X., and Le, T. "Construction of B2B Electronic Commerce System Based on Apache Struts Framework," International Conference on Services Science, Management and Engineering, 2009, pp 221-224.

Struts 有以下優點：

- 可以很容易地開發大型 Web 應用程式。Struts 是 open-source 框架，Struts 切成 JSP、ActionServlet、Action 使得 MVC 架構更清楚明瞭。Action 簡化了商業邏輯，複雜的商業邏輯則由 JavaBean 和 EJB 做處理。因此開發複雜系統的開

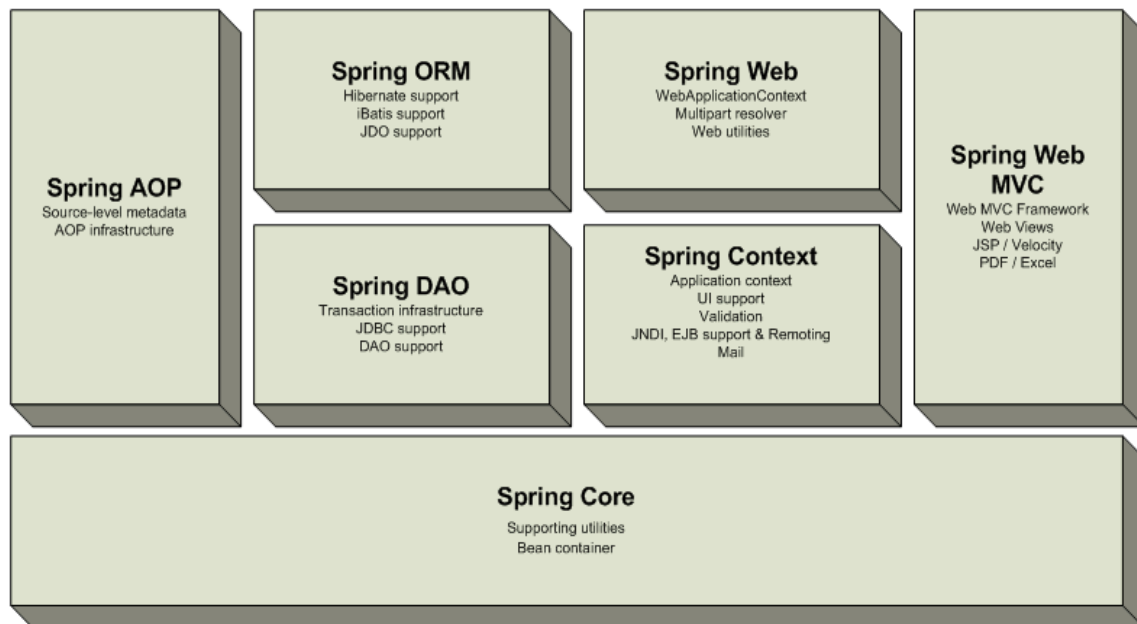
發被簡化了。

- 透過 XML 和 property 檔來配置，而非寫在 Java 程式碼裡。這樣在面對變更時，不需要大量的修改原程式或重新編譯。大批的修改也只需要透過單一檔案就可完成。這個方法使得開發者可以更專注在某些特別的任務上(處理商業邏輯)而不需要去知道關於所有系統的設計布局。
- Struts 包含了強大的 JSP 標籤函式庫，像是 Bean Tags、HTML Tags 並且支援第三方函式庫。
- 驗證欄位形式用更容易的方式來達到驗證錯誤的功能，像是如果表單裡面沒填值，則自動地填入原本預設的值進去或著顯示出錯誤訊息。

### 三、Spring 框架

Spring Framework 是一個輕量級 J2EE 應用開發框架，由 DAO、ORM、AOP、Web、Context、Web MVC、Core 數個模組組合而成。它使用 IoC(Inversion of Control)來分離實際程式的配置和相依性規則，把所有層級的解決方式交給 J2EE 應用程式。Spring 並不是企圖放一個已存在的框架，而是透過 POJO(Plain Ordinary Java Object)管理來對所有層級的 J2EE 應用做一個融合。除此之外，開發者可以自由選擇要使用所有的 Spring 框架或其中一部分，因為 Spring 子模組互相並不相依。身為主要負責商業邏輯層級細節，Spring 使用 delay injection 的想法來組合程式碼以此來增進建立系統的擴增性和彈性，因此系統能達到中央集權的商業邏輯處理和透過 Spring AOP 模組來減少程式碼的重複性(Liu et al. 2008)。





Overview of the the Spring Framework

圖 2-7 Spring 框架

資料來源：Spring 官方網站，2012，<http://www.springsource.org/>

各模組功能如下：

1. **Spring Core**： 整個 Spring 的核心，管理物件的生命週期、物件組態以及相依注入，主要在實作工廠模式(Factory Method)並實現 IOC 功能，透過配置檔將物件關係與實際的程式碼切割開來。
2. **Spring AOP**： 為了重複利用程式碼的一種作法，Spring 大部分的 AOP 是基於 Alliance 的 API 開發的，通過程式碼中的 Annotation 來簡化 AOP 配置工作，對所有以 Spring 為基礎之物件提供了宣告式的事務管理服務。
3. **Spring DAO**： 資料庫連接的介面，對於 JDBC 進行抽象與封裝，透過 JDBCTemplate 大量簡化資料庫操作，減少無意義的程式碼，並提供有意義的例外處理架構，可以用來處理不同資料庫供應商拋出的錯誤訊息。
4. **Spring ORM**： Spring 對於現今各大 ORM 框架均提供良好的封裝與整合，例如像是 Hibernate、JDO、iBatis。

5. Spring Web: Spring 自己有提供 MVC 框架，也提供了其他框架的整合像是 Struts 框架整合。
6. Spring Context: 為一個配置檔，向 Spring 核心提供如 JNDI、EJB、驗證、郵件以及國際化等功能。
7. Spring Web MVC: 全功能的 Web 框架，採用大量的呈現技術，包括 JSP、Velocity、Tiles、iText 和 POI。

Spring 有以下優點：(林信良 2008)

- 輕量級(Lightweight): Spring 的核心在檔案容量上只有不到 1MB 的大小，而使用 Spring 核心所需要的資源也是很小的。
- 容器(Container): 控制物件在創建時是以原型(Prototype)或單例(Singleton) 的方式來建立。
- IoC(Inversion of Control): 不必自己在程式碼中維護物件的依賴關係，只需在組態檔中加以設定，Spring 核心容器會自動根據組態將依賴注入指定的物件。
- 非侵入性(Nonintrusive): 讓應用程式不感受到框架的存在，減低應用程式從框架移植時的負擔。

#### 四、Hibernate 框架

Hibernate Framework 是現今最普及的 ORM(Object Relation Mapping)框架。它是當作 Java 物件和關聯式資料之間的橋樑，也對 JDBC 做了封裝。它不僅提供 ORM 的映對服務，還提供資料查詢和資料緩衝服務。Hibernate 封裝 SQL 的操作，使得開發者可以方便的透過 Hibernate API 存取資料庫。在 Java Web-Based 應用程式中，Hibernate 主要用在資料存取層。

Hibernate 透過 configuration files(hibernate.cfg.xml)和各個資料表的 mapping files(\*.hbm.xml)來對 Java 物件進行映對，然後透過操作 PO(Persistent Object)，我們可以

在物件導向程式中對資料庫中的資料進行新增、刪除、修改以及查詢等操作(Shan and Yuntao 2011)。

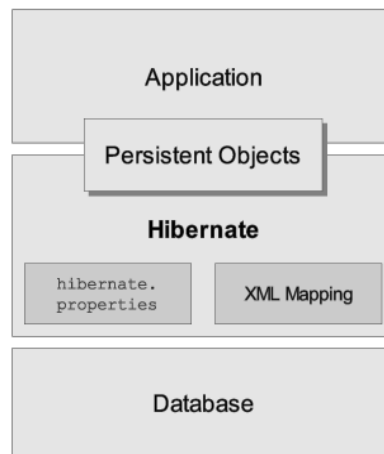


圖 2-8 Hibernate 框架

資料來源: Hibernate 官方網站，2012，<http://www.hibernate.org/>

Hibernate 有以下優點：

- 支持許多現今的資料庫供應商。
- 使應用程式開發可以只操控物件，完全以物件導向為中心，不須花費額外力氣去管理資料庫。
- 具有優秀的移植性，移植僅需要更換配置檔。
- 提供簡潔以物件為導向的 HQL 語言
- 無依賴於特定容器

## 五、SSH 整合(Liu 2011)

### 1. Spring 與 Struts 整合

當這兩框架整併時，Struts 的 action 會授權給 Spring。利用 Spring 供應的 DelegatingActionProxy 類別，將 request 送給那些在 Struts 配置檔(struts-config.xml)裡的實際 action。把 DelegatingActionProxy 類別當作 action 的 proxy，這樣做可以讓 Spring 來管理由 Struts 所呼叫的 action 實例。

## 2. Spring 與 Hibernate 整合

Hibernate 最重要的配置檔是 hibernate.cfg.xml，所有需要持久化的類別和相關的全域屬性都被定義在這檔案裡。當這兩個框架整合時，Hibernate 將被 Spring 管理，也就是 hibernate.cfg.xml 被 Spring 管理。Spring 藉著 LocalSessionFactoryBean 類別來執行 Hibernate 的介面轉換，使我們可以用一致的方法來處理 Hibernate 和 Spring。在 Hibernate 的應用程式中，Spring 的類別 Hibernate Template 可以完整的作用來當作捕捉例外、管理交易等等而不需要其他任何多餘的程式碼。

## 3. 三者整合

在 Spring 分別和 Struts 和 Hibernate 整合後，接著複製 jar 套件到類別路徑中，則三個框架即整合完成。下圖為整合三個框架，每個框架所負責之工作結構 (Yongchang et al. 2011)。

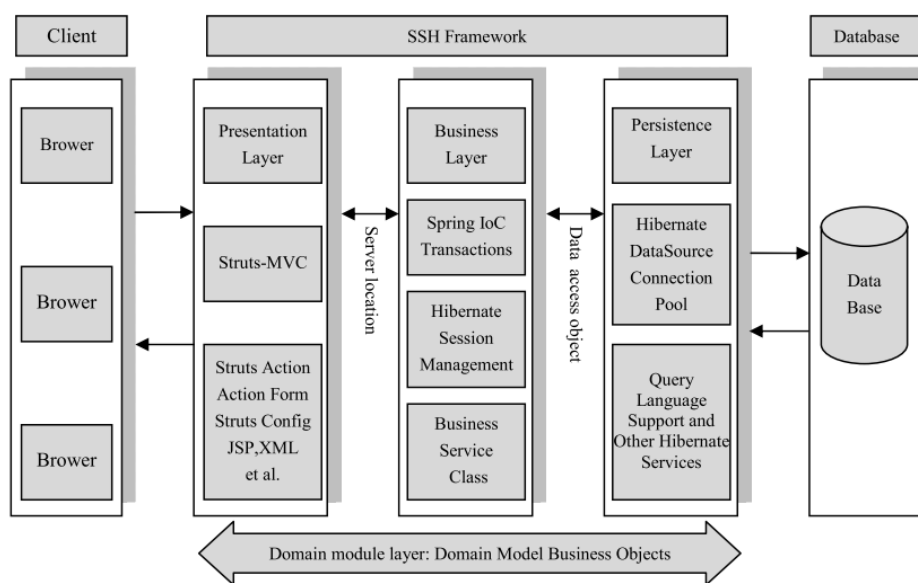


圖 2-9 SSH 整合框架結構圖

資料來源: Yongchang, R., Xing, T., Xing, Z., and Zheng, J. "Application Research for Integrated SSH Combination Framework to Achieve MVC Mode," Computational and Information Sciences, International Conference on (0) 2011, pp 499-502.

下圖展示 SSH 整合框架資料傳遞流程：

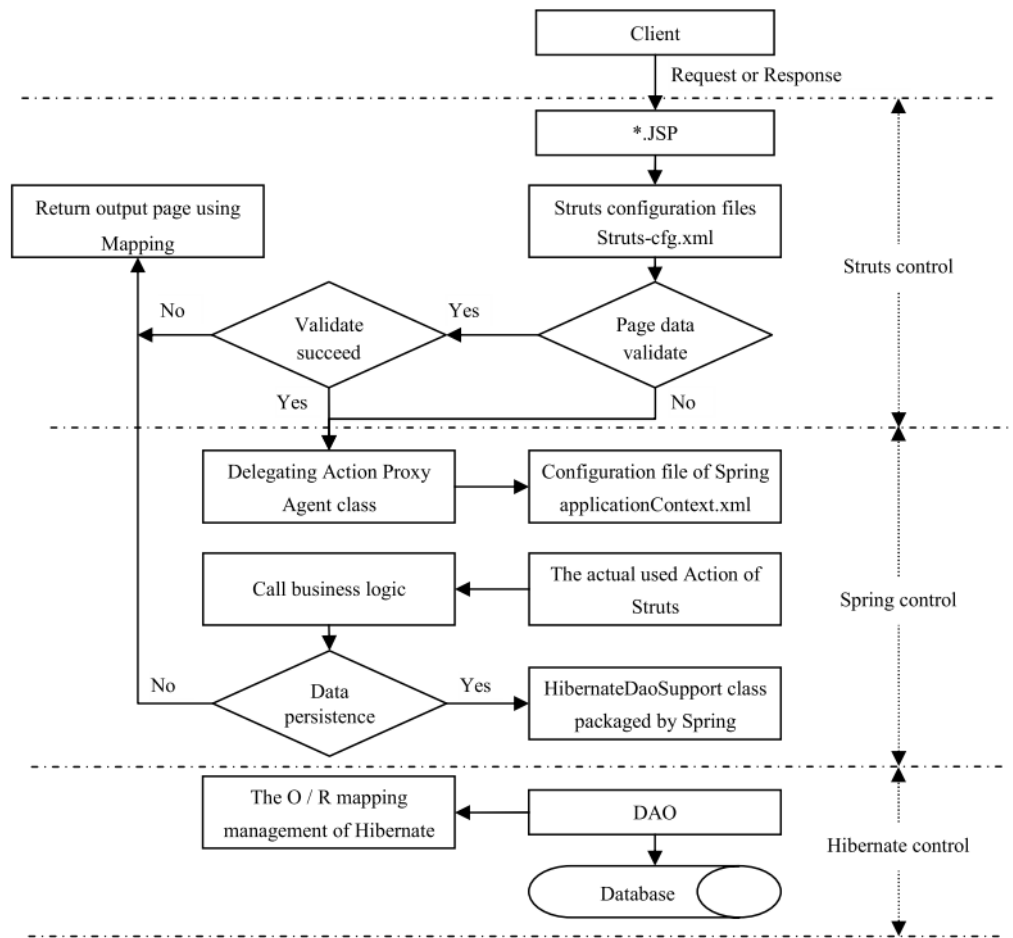


圖 2-10 SSH 整合框架流程圖

資料來源: Yongchang, R., Xing, T., Xing, Z., and Zheng, J. "Application Research for Integrated SSH Combination Framework to Achieve MVC Mode," Computational and Information Sciences, International Conference on (0) 2011, pp 499-502.

### 第三節 UML

UML 是軟體發展人員為了提升團隊工作的溝通效率，將系統模型以具現化方式建造。UML 由三位物件導向方法論專 Grady Booch、James Rumbaugh 及 Ivar Jacobson 共同合作制定的物件導向塑模語言。根據物件管理組織( Object Management Group ，簡稱為 OMG )的定義， UML 是「用來詳細說明、呈現、建置、記錄軟體系統工作成果的一種塑模語言，它也可以用來建立企業模型以及其它非軟體系統。」(Object Management Group 2001)

#### 一、UML 架構(葉文崎，2002)

##### 1. 使用案例觀點(Use Case View)

可說是 UML 的核心，顯示外界的行動者(Actioners)所使用到的系統功能。這個觀點並不說明一個軟體系統內部的構造，而是純粹地解說系統與行動者間的互動。

##### 2. 邏輯觀點(Logical View)

這個觀點主要顯現出系統內部如何透過物件類別的責任分配與互動運作以提供系統功能，提供系統設計開發者使用。

##### 3. 實作觀點(Implementation View)

相關的一組類別經編譯之後產生可執行的軟體元件，其間的互動說明系統內部各模組元件的資訊及彼此之間的相依性(Dependency)。

##### 4. 程序觀點(Process View)

在這個觀點中考量的重點就是如何透過對執行程序，或是執行緒(Thread)的組態(Configuration)，來完成系統功能性的需求。同時能符合系統中非功能性(Nonfunctional)的屬性。

##### 5. 部署觀點( Deployment View)

這個觀點主要顯示出系統軟硬體架構的實體佈署。

## 二、UML 圖形

表 2-7 UML 圖形分類

靜態模型 (Static Models)	使用案例圖 (Use Case Diagram)
	物件圖 (Object Diagram)
	類別圖 (Class Diagram)
	元件圖 (Component Diagram)
	佈署圖 (Deployment Diagram)
動態模型 (Dynamic Models)	活動圖 (Activity Diagram)
	狀態圖 (Statechart Diagram)
	循序圖 (Sequence Diagram)
	合作圖 (Collaboration Diagram)

資料來源: 葉文崎, 2002, 元件式系統開發方法研究—以郵務應用為例, 國立政治大學資訊管理研究所碩士論文。

### 1. 使用案例圖(Use Case Diagram)

用以描繪出系統所包含的使用案例(Use Case), 以及與系統外部角色間的互動關係, 並以 Use Case 作為所有軟體開發流程的起始點。

### 2. 活動圖(Activity Diagram)

透過對活動工作者(workers)、工作流程(workflow)、組織(organization)及相關物件的互動, 活動圖可用來表達一個企業流程內部如何運作。

### 3. 類別圖(Class Diagram)

類別圖可用來描述系統中物件的類型(types)及各類型間的靜態關係。類別包含了類別名稱(name)、內部屬性(attributes)及外部的操作方法(operations)。

### 4. 狀態圖

狀態圖對於系統行為模型的描述非常有用, 它可以把一個物件可以進入的所有可能狀態, 以及這物件在面對外部事件所進行的狀態變化情形全都表示出來。

## 5. 循序圖(Sequence Diagram)

循序圖是互動的一種，主要強調物件間訊息傳遞的時間順序，循序圖和合作圖主要有兩點差別，循序圖有生命線(Lifeline)及控制焦點(Focus of Control)，生命線表現出物件互動的時間；控制焦點表示物件處理某項工作的所耗費的時間。

## 6. 合作圖(Collaboration Diagram)

合作圖亦是互動圖的一種，主要強調物件間如何參與互動。合作圖和循序圖的差異主要在於，合作圖的空間配置較自由，但無訊息傳遞時間順序，不過此缺點可由順序編號(sequence number)來彌補，透過順序編號可以清楚的知道物件互動的順序。

## 7. 物件圖(Object Diagram)

物件圖可表達一組實體物件間在某一時點的關係。

## 8. 元件圖(Component Diagram)

元件圖最主要表示出當系統開發完成後的軟體執行架構，以及元件彼此之間的相依關係(dependency)，一般會將之與佈署圖結合作為系統之軟硬體實體架構模型。

## 9. 佈署圖(Deployment Diagram)

配合元件圖，佈署圖可以把系統的軟體元件、硬體設備之間的關係表示出來，描述分散式系統中所包含的硬體設備種類及軟體元件在其中的具體分布情形。



#### 第四節 人力資源

人力資源管理指的是企業對於人力資源做相對應的管理活動，這些活動包括員工招募、員工培訓、績效管理、薪酬管理、員工流動管理、員工關係管理等。眾多學者提出各種不同的看法，本研究依照薛昊平(2010)論文研究所提到對於人力資源管理的彙整理出如下表：

表 2-8 人力資源管理理論

學者	年代	理論內容
吳秉恩	1992	確認企業整體的發展目標、掌握市場機會、顧及企業資源條件三大原則下，發揮人力資源效率。
黃英忠	1992	將組織內所有人力資源做最適當之確保、開發、維持與活用。
Becker & Gerhart	1996	主張組織的人力資源策略須與策略規劃的程序與組織特徵相互配合。若透過人力資源管理活動培養出組織所希望的員工特質、技術與行為，則組織策略成功的機率將會增加。
簡建忠	1997	人力資源在企業組織中經由規劃、篩選、工作和組織設計而扮演系統創造者的角色;透過勞資關係、員工參與、訊息交流和激勵措施而扮演系統改善者的角色;憑藉員工職涯發展;組織發展和績效改進而扮演系統改善者的角色。
Mathis & Jackson	1977	人力資源管理的目標主要包括生產力、品質及服務三部分。
林財丁與陳子良	2002	人力資源管理具有策略性競爭優勢，當其與企業整體策略相結合時，也扮演著釐清公司人力資源問題、發展解決方案的重要角色。
Ulrich & Brockbank	2005	將人力資源區分為策略夥伴、人力資本發展者、員工支持者、功能性專家及人力資源領導者五種角色。
Collins	2005	卓越企業的策略是不斷引進人才，為公司注入新血，並且淘汰不適合的人，企業始能開創未來，找到新方向、新願景和新策略。
紀乃文、張火燦	2006	人力資源主管之策略夥伴角色對企業之經營績效有顯著正向影響，而策略夥伴角色會透過員工關係指標間接影響組織績效。

資料來源：薛昊平，2010，人力資源管理系統導入之個案研究—以 C 公司為例，國立中央大學資訊管理研究所碩士論文。

## 一、人力資源資訊系統(HRIS, Human Resource Information System)

資訊時代的來臨，人力資源的管理也逐漸電子化，企業從過去導入 ERP、SCM、CRM 等系統，對於企業內外部資源做一個統整，也逐漸發現當企業規模變大，企業的人才管理開始出現困難，過去的紙本記錄容易出錯且沒效率，因此開始導入人力資源資訊系統(HRIS, Human Resource Information System)來處理企業組織內與人力資源相關的資訊與流程，運用 HRIS 可以系統化分析人力資源資訊，進行組織中完整的資訊資源管理，進一步可以提高人力資源管理部門的營運績效。

在人力資源資訊系統中會分為許多子系統，子系統為完整資訊系統中執行單一任務的模組或程式，關於這方面國內外學者有不同看法，本研究參考方麗玉(2011)論文所彙整之人力資源資訊系統所應有模組及應用。本研究將依照 Mcleod, Raymond Jr. and Desqunctis Gerardine 學者所提出的 HIRS 系統模組，來建造系統模型。

表 2-9 人力資源資訊系統

學者	年代	HIRS 子系統
Kavanagh, Gueutal & Tannenbaum	1990	基本人事資料模組、應徵者追蹤模組、招募模組、績效管理模組、生涯發展模組、職位控制模組、薪資模組、福利模組、公平就業模組、訓練模組、人力資源規劃模組。
Beutell & Walker	1993	應徵者追蹤與招募、用人、員工統計訓練與發展、薪資、福利、退休金計算保險。
何永福、楊國安	1993	人力資源資料庫的建立、薪資作業、福利管理、員工人力規劃、人力資源作業報告。
Mcleod, Raymond Jr. and Desqunctis Gerardine	1995	人力規劃子系統、招募子系統、人力管理子系統 (Workforce Management Applications)、薪酬子系統、企業環境報告子系統(Environmental Reporting Applications)。

資料來源：方麗玉，2011，國內企業導入人力資源管理資訊系統之研究，國立中央大學人力資源管理研究所碩士論文。

### 第三章 系統發展研究架構

目前在以 Scrum 敏捷開發流程為主的開發方式中，對於屬於系統高階規劃的軟體架構均無明確的作法，本研究藉由以往相關文獻的探討，以及相關實務經驗，提出一套融合了 Scrum 和 Struts、Spring、Hibernate 整合式框架為基礎的系統開發方法。將以開發 Java Web-Based 人力資源系統雛型為例，也就是以 Scrum 為開發流程基礎，利用整合式框架作為系統骨幹，建立一套應用軟體開發程序。

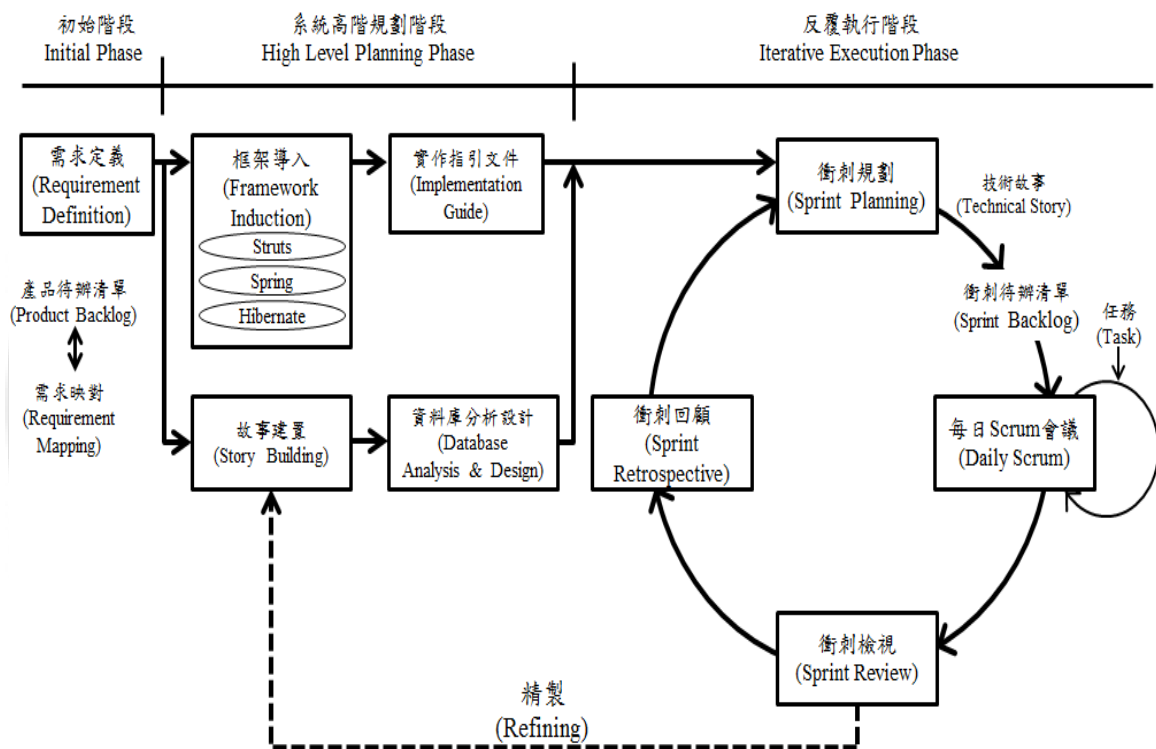


圖 3-1 系統開發程序及方法

資料來源：本研究整理

## 第一節 初始階段(Initial Phase)

此階段為找出所要開發系統的目標及效益並規範出可收斂的專案範圍。確定目標及範疇後，必須取得需求，取得需求的方式可以透過和客戶訪談的方式，將之紀錄在會議訪談記錄中，再由訪談結果得出客戶所希望的功能性需求或其他需求，本研究以 UML 中的使用者案例圖所取得的需求建模。

### 一、需求定義(Requirements Definition)

由產品負責人負責，先透過與客戶溝通了解客戶需求，利用使用者案例圖先將所獲取的需求建模，在 Scrum 中把定義出的系統功能性需求稱為故事，而將故事集成表建立產品待辦清單。產品待辦清單是 Scrum 的核心，由故事依重要順序排序而成，故事在 Scrum 中指的是以客戶的術語，來描述想要的東西(Kniberg 2007)。

### 二、需求映對(Requirements Mapping)

由產品待辦清單上所列之故事項目和之前舊有的系統元件做比對，其來源可能有三種，分別是元件市場上取得的相關領域軟體元件、相關領域再用軟體框架以及組織內部元件程序的產出(葉文崎 2002)，若有相同的項目則不需要重複浪費工時開發，從產品待辦清單上去除。若有相似的項目，則可以直接取來做修改，以增加開發效率，本階段可製作映對結果描述文件來做檢視。

## 第二節 系統高階規劃階段(High Level Planning Phase)

### 一、 框架導入(Framework Induction)

選擇適當的技術框架導入，本研究以 Java 語言為主要開發語言，由於目前許多服務已走向 web 化，下圖為在 Java Web 框架的使用普及程度，而本研究將採利用 Spring 框架當核心，透過配置檔案，將 Struts 和 Hibernate 框架整合。

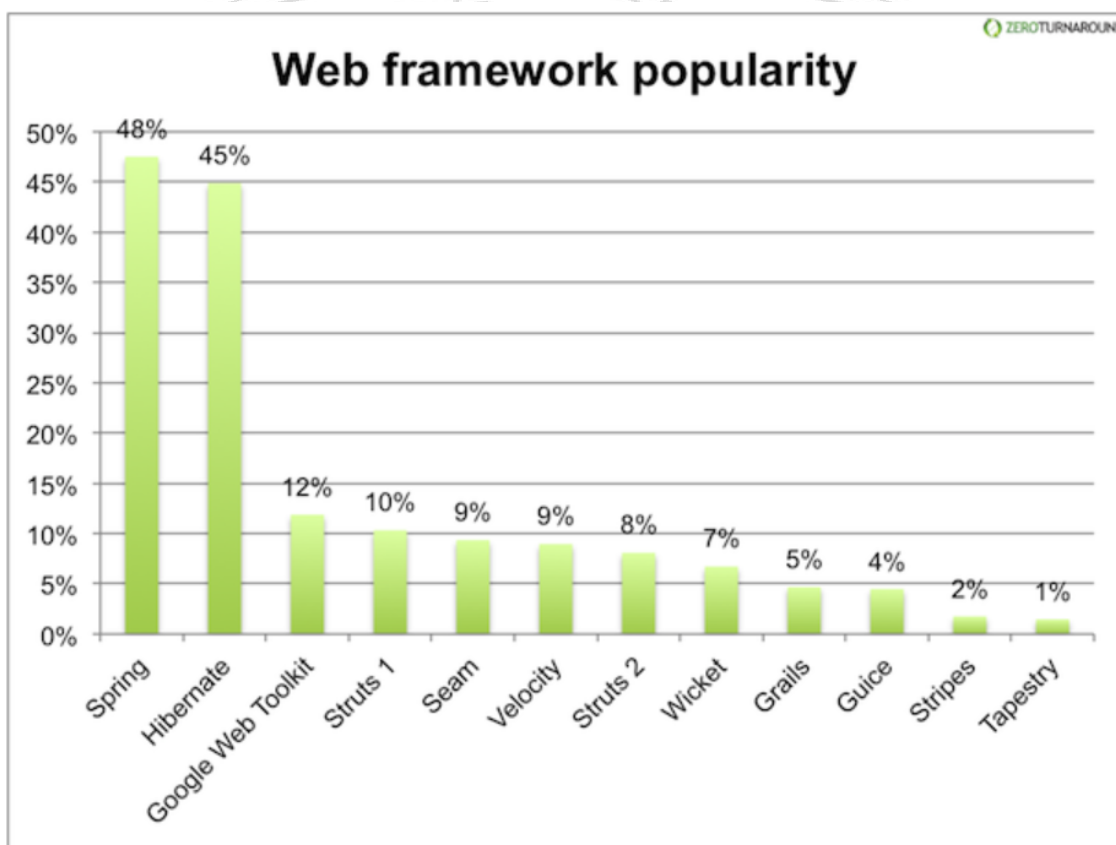


圖 3-2 Java 普及框架比例圖

資料來源: ZeroTurnaround, 2011, <http://zeroturnaround.com/java-ee-productivity-report-2011/>

## 二、故事建置(Story Building)

產品擁有者以 top-down 的方式，將整體系統概觀的主流程透過活動圖畫出。再透過簡短的描述句子來對每一項故事做一個簡單的敘述，使團隊成員可更容易了解每一項故事所代表之涵義，如下表所示：

表 3-1 故事描述表

ID	Name	Description
1	登入	登入畫面包含讓使用者輸入使用者帳號及密碼，並且有驗證處理。

資料來源:本研究整理

## 三、技術功能展開(Technical Function Expansion)

Scrum Master 透過整合框架結構圖以及循序圖依循架構分層負責的概念，參考張弘明(2007)所提出之 DFE 文件法，加上開發團隊需要一致維持之技術規範，建立實作指引文件(Implementation Guide)，並要求團隊成員共同遵守。實作指引文件包括以下項目：

### 1. 基礎設施(Infrastructure)

包含框架結構設計、資料庫、應用伺服器的概觀圖以及 package 的分類表。下表為 package 分類表的範例：

表 3-2 package 範例分類表

Package	說明
Src	Source 相關
<b>HRSsystem_SSH2.edu.nccu.hr.common.*</b>	將共用部分抽象出來當元件以利其他專案重複使用。

資料來源:本研究整理

### 2. 整體循序圖(Overall Sequence Diagram)

以循序圖來描述框架分層結構，使專案團隊人員更容易了解分層結構，並可以統一分層規格。

### 3. 開發平台(Developer Platform)

包含開發文件存放位置以及程式碼目錄結構的敘述。

### 4. 開發環境(Development Environment)

開發環境的規格制訂，使開發團隊可以統一版本的軟體進行開發，減少版本不一致導致的麻煩，像是資料庫、應用伺服器、JRE(Java Runtime Environment)、開發環境、正式環境以及 IDE(Integrated Development Environment)等。

### 5. 開發工具(Developer's Tools)

像是程式的開發工具、技術上使用到的框架以及文件撰寫的工具等，使開發團隊可以使用相同版本工具做開發，加快開發速度。

### 6. 共用函式(Common Library)

像整合框架會導入許多函式庫，常常會發生函式庫互相的衝突，所以將共用的函式庫記錄下來，當專案需要轉移或交付的時候，就可解決函式庫衝突問題。

## 四、資料庫分析與設計(Database Analysis and Design)

Scrum Master 依據故事建置以及需求，進行設計整體系統的資料庫架構。欄位設計必須進行關聯鍵值設計以及正規化作業，並依此產生實體關聯圖(Entity Relationship Diagram, ERD)，使團隊成員可了解資料庫中各實體(entity)相互關係，並製作欄位說明表單，以讓團隊成員皆能快速了解每個表格中欄位所代表的意義。

設計出資料庫各實體之間的關聯關係後，便可依此製作相關的程式類別圖，以圖示方式表達：

資料來源：本研究整理

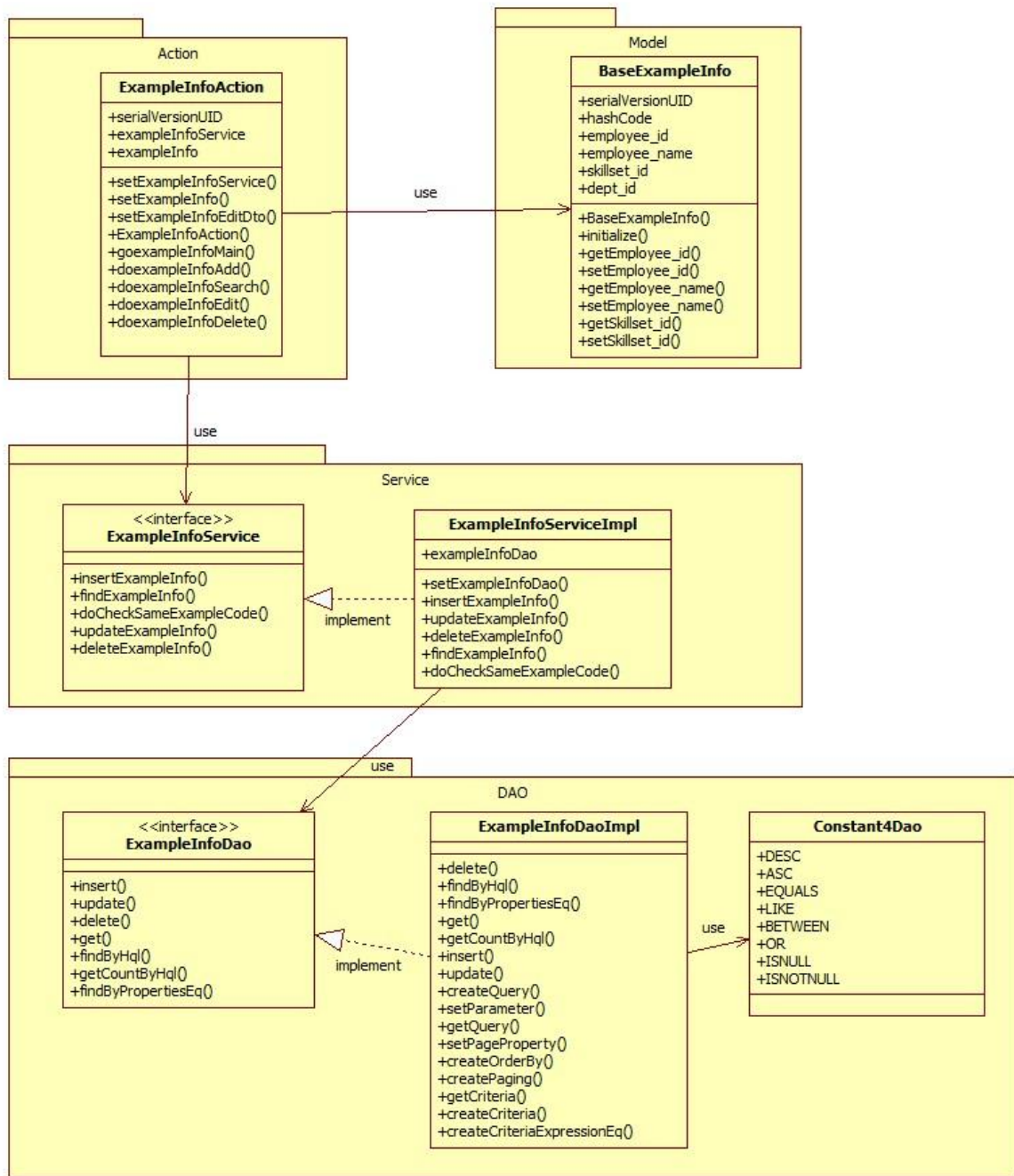


圖 3-3 類別範例圖

資料來源：本研究整理



## 反覆執行階段(Iterative Execution Phase)

### 一、衝刺規劃(Sprint Planning)

由產品負責人決定此次衝刺中所要達到的目標，還有他認為最重要的故事。然後團隊開始討論每個故事，並估算時間，最後確定衝刺週期長度與衝刺所含的故事，並產出衝刺待辦清單。衝刺規劃應該具有以下具體成果：

1. 衝刺目標
2. 團隊成員名單
3. 衝刺待辦清單
4. 衝刺確定的展示日期
5. 每日 Scrum 會議所舉行的固定時間與地點

由於 Scrum 中一切事情都是有時間盒(time box)，所以衝刺規劃的會議不應該無止盡的討論，而是有個固定的時間限制，Scrum Master 應決定衝刺規劃會議的固定長度，若討論不出結果，則先中止此次會議，以目前討論結果去進行此次衝刺。

### 二、技術故事(Technical Story)

依據 Kniberg(2007)所定義，技術故事指的是需要完成但又不屬於可交付的東西，和任何故事都沒有直接關係，不會為產品負責人帶來直接的價值。以下為技術故事舉例：

1. 安裝持續整合伺服器

每日自動將團隊所交付的成果整合進某一台開發機器中，交由測試人員測試，這樣在最後展示的時候較不易出現統整的重大問題。

2. 使用版本管理伺服器

透過版本管理使團隊開發的程式碼版本一致。

3. 重構程式碼

由於團隊一起開發常會出現複製程式碼的情形，時間一久就會出現許多重複的

程式碼，又或是程式碼中出現許多缺失，此時應該花時間進程式碼重構，以利接續的開發進行。

#### 4. 軟體版本更新

軟體可能出現新版本，比現有版本少許多臭蟲(Bug)，或是更方便使用，這時候就可以對所使用的軟體進行版本更新。

技術故事由於對於可交付成果沒辦法看到立即的成效，且會花額外的時間，但對於未來開發可以提升大幅度效率，應由 Scrum Master 向產品負責人提議，由產品負責人決定是否採用技術故事。

### 三、每日 Scrum 會議(Daily Scrum Meeting)

每日同時間同地點短期開會 15 分鐘，由 Scrum Master 向團隊成員提問三個問題分別是：從上次會議到現在完成了哪些工作、下次每日 Scrum 會議前準備完成什麼以及工作中遇到那些阻礙。

在每次每日會議時，每個團隊成員會一邊描述已經做的事和今天要做的事，一邊移動任務版上對應的便利貼，可以減少 Scrum Master 花太多時間在管理的工作上，再來便是如果團隊成員不自己更新任務狀態，則對任務敏感度與集中度皆會下降，本研究採用電子化的方式使用 ezScrum 來做管理，一方面讓團隊成員可用上網的方式改變任務狀態，另一方面的好處則是減少紙張的浪費。

### 四、任務(Task)

依據 Kniberg(2007)所定義任務指的是將故事細分出許多部份，每個部份皆稱做任務，與故事的差別是任務是不可交付的東西，產品負責人並不關心任務，一般會在衝刺規劃會議最後的剩餘時間進行故事的拆分。

### 五、衝刺檢視(Sprint Review)

在衝刺結束時召開衝刺檢視會議，在這時做衝刺成果展示，由產品負責人確定哪些工作已完成，那些工作未完成。最後產品負責人和團隊成員討論並按目前進度推測可能需要的時程，並評估產品待辦清單所需的修改。同時若產品負責人覺得需要額外加一些功能，也可能再去調整故事建置。以下為衝刺檢視檢查項目：

1. 確保清晰闡述衝刺目標，如果展示中有人對產品展示不了解或不清楚，則要對產品進行描述。
2. 不需花太多時間準備檢視，不需要做太過花俏的簡報，應該集中精力於展示程式碼。
3. 節奏要快，不需要過於繁冗的講述，只要大家都清楚明瞭產品的展示就可以。
4. 不管技術細節，由於檢視會議中還包括了許多利害關係人，有些人對於技術並不十分了解，他們關注的是團隊做了什麼，而非團隊如何做出來這個產品。
5. 不要演示一大堆微不足道的特性及臭蟲(Bug)，這樣會分散大家的注意力，應該把重點擺在展示重要的故事上面。
6. 可能的話交由與會者操作產品。

## 六、精製(Refining)

當衝刺檢視結束，產品負責人可能發現需要多加功能或修改產品項目，則透過修改產品待辦清單來達成，但這不代表產品負責人可隨時更改產品待辦清單，每次衝刺開始後，衝刺目標是不可改變除非中止整個衝刺，唯有等到衝刺檢視結束才可進行功能新增或修改。

## 七、衝刺回顧(Sprint Retrospective)

Scrum Master 和團隊成員討論前一個衝刺需要調整或改善的地方，可對下一個所要實施的衝刺進行修正。

依據 Kniberg(2007)指出通常衝刺回顧會議會有以下討論事項：

1. 固定時間長度為 1 至 3 小時，內容範圍為上一個衝刺執行。
2. 對上一個衝刺做結論，包括重要事項和決策。
3. 每個人輪流發言，講出自己的想法，看如何做可改善，哪些是下一個衝刺需要改變的，當每個人發言時其他人不可打斷。
4. 對預估生產率和實際生產率進行比較，差異過大則分析原因。



## 第四章 系統發展實作-以人力資源管理系統雛型為例

本研究以人力資源管理系統雛型為例，利用本研究提出的三階段架構，執行系統實作，框架選擇使用 Java Web 框架 Struts、Spring、Hibernate，並搭配 JavaScript 框架 JQuery 達到 AJAX(Asynchronous JavaScript and XML)效果，使前端使用者有速度提升的感覺。Scrum 為一個成熟的敏捷開發框架，非常適合需求變更快速的環境，整體框架也十分有彈性，是一個以人為中心的專案管理方式，但也由於如此的彈性，對於技術方面 Scrum 只有粗略地提及，所以當面對像本研究採用結合多個框架的較複雜開發專案，有可能導致專案後期窒礙難行，因此本研究以人力資源管理系統雛形，探討一個較為完整的系統開發程序。

### 第一節 初始階段

#### 一、需求定義

本研究所開發的人力資源管理系統雛形，主要定位於人力派遣以及課程開班對於人力需求的使用，以下為所定義出之系統細部項目：

##### 1. 代碼管理

由系統管理員進行維護，對於資料類型之代碼給予一定的命名規則，像是客戶代碼、技能代碼等。子項目則有代碼基本資料維護。

##### 2. 權限管理

由系統管理員進行維護，對於使用系統角色給予不同權限，例如將系統利害關係人分為人力資源管理員、課程管理員，不同權限所可檢視之子系統項目不盡相同。子項目之授權使用者設定於權限基本資料維護。

##### 3. 人員管理

人員管理所維護的為人員基本資料的維護，可由人員本身或是人力資源管理員進行維護更新。子項目則有人員基本資料維護、人員學歷資料維護、人員經歷資料

維護、人員訓練資料維護、人員技能資料維護以及人員聯絡資料維護。

#### 4. 課程管理

對於所開班的課程資料進行維護，由課程管理員負責，子項目則有課程套裝資料維護、課程基本資料維護、課程開班資料維護以及課程聯絡資料維護。

#### 5. 人力需求管理

人力資源管理員在取得所需人力需求後所進行的資料維護項目。子項目則有人力需求資料維護、人力需求明細維護、客戶聯絡人資料維護以及人力需求聯絡維護。

#### 6. 上課管理

課程管理員對於人員上課情況所進行之資料維護。子項目則有人員上課資料維護。

#### 7. 工作管理

此項目為人員自行對工作的簽到簽退資料進行維護，再由上級主管對此項目進行查核動作。子項目則有人員工作資料維護。

#### 8. 需求建模

本系統的使用者角色可分為五種角色分別為人員、人力資源管理員、課程管理員、系統管理員以及人員上級主管，各種角色對於系統底下的子系統有不同對應的互動關係，此時由產品負責人依據使用者角色與系統互動之關係描繪出使用者案例圖。

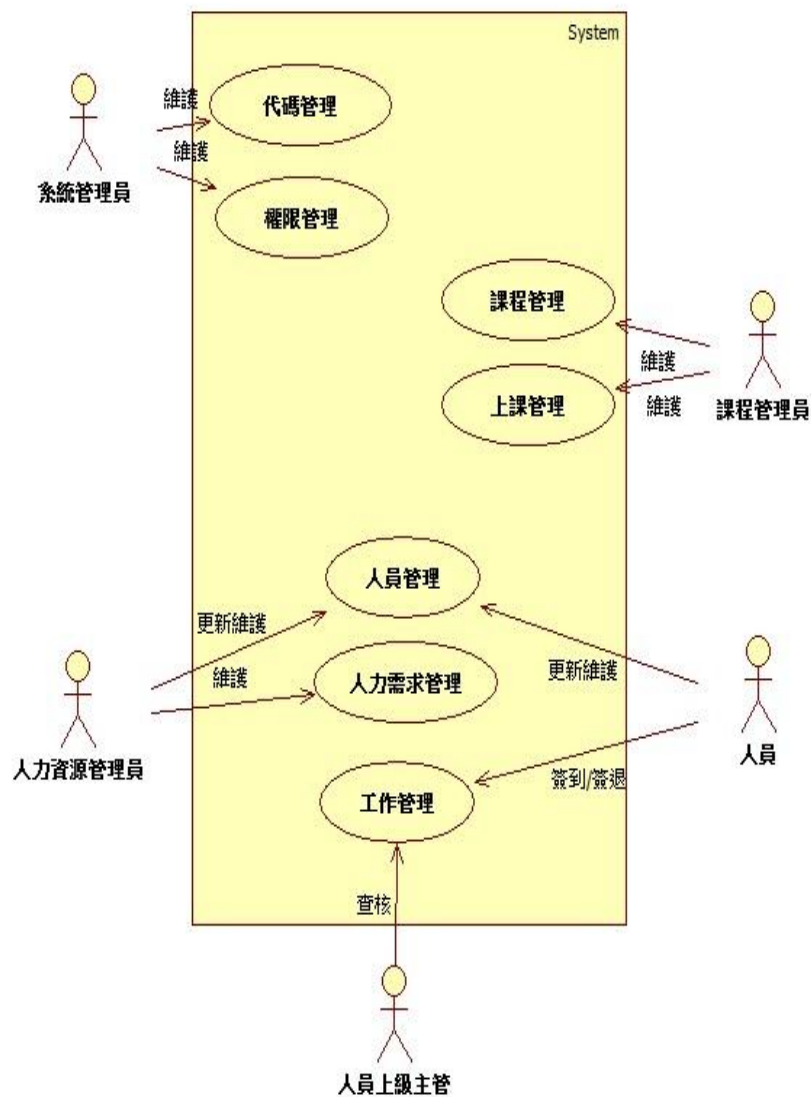


圖 4-1 系統使用者案例圖

資料來源：本研究整理

## 9. 產品待辦清單

產品負責人依據使用者案例圖，考慮產生系統應該具備的故事，並製成產品待辦清單，每個故事包含了幾個項目分別是 ID、故事名稱、發行版本、衝刺、價值、初始估算、重要性、如何演示以及註解。

- ID：每個故事都有個唯一的識別 ID，提升找尋故事的速度。
- 故事名稱：對於故事一個簡短性的描述名稱，一般由 2~10 個字組成。
- 發行版本:標示此故事是屬於專案中的哪個時期發行版本。

- 衝刺：標示此故事是屬於哪個衝刺需要完成的事項。
- 初始估算：團隊的初始估算，最小的單位是故事點(story point)，一般相當於一個理想的人天(man-day)。
- 重要性：由產品負責人評估出此故事的重要性並給予一個數值以表示重要程度，分數越高越重要。
- 如何演示：簡略的敘述這個故事該如何在衝刺檢視上進行演示。
- 註解：對於故事的相關訊息、解釋說明以及對於其他資料的引用等。

下圖為編輯一項故事的範例：

The screenshot shows a dialog box titled "Edit Story" with the following fields and content:

- ID:** 19
- Name:** 人員基本資料維護
- Value:** 80
- Estimation:** 12
- Importance:** 80
- Notes:** CRUD功能需使用AJAX技術作資料間的傳遞
- Tags:** (empty dropdown menu)
- How To Demo:** 點擊新增可寫入一筆新的資料進資料庫；點擊刪除可將所選擇之資料刪除；對所顯示的資料點擊可進行資料細部項目修改；查詢可依據使用者所希望輸入的條件，進行動態條件查詢。

At the bottom right of the dialog, there are two buttons: "Submit" and "Cancel".

圖 4-2 系統故事

資料來源：本研究整理



所產生之產品待辦清單：

Id	Tag	Name	Release	Sprint	Value	Estimation	Importance	Status
19		人員基本資料維護	None	7	80	12	80	new
38		人員工作資料維護	None	None	0	0	0	new
17		權限基本資料維護	None	None	0	0	0	new
16		權限管理	None	None	0	0	0	new
15		新增人力需求	None	None	0	0	0	new
14		代碼基本資料維護	None	None	0	0	0	new
13		代碼管理	None	None	0	0	0	new
20		人員學歷資料維護	None	None	0	0	0	new
21		人員經歷資料維護	None	None	0	0	0	new
22		人員訓練資料維護	None	None	0	0	0	new
23		人員技能資料維護	None	None	0	0	0	new
24		人員聯絡資料維護	None	None	0	0	0	new
25		課程管理	None	None	0	0	0	new
18		人員管理	None	None	0	0	0	new
27		課程基本資料維護	None	None	0	0	0	new
28		課程開班資料維護	None	None	0	0	0	new
29		課程聯絡資料維護	None	None	0	0	0	new
30		人力需求管理	None	None	0	0	0	new
31		人力需求資料維護	None	None	0	0	0	new
32		人力需求明確維護	None	None	0	0	0	new
33		客戶聯絡人資料維護	None	None	0	0	0	new

圖 4-3 系統產品待辦清單

資料來源：本研究整理

## 二、需求映對

產品負責人透過和過去所做的專案產生元件比對，若有相同或相似之功能則將該功能自產品待辦清單移除，但由於本研究此次所做之人力資源管理系統雛型，主要為技術面改版，由於框架版本不同，使用差異大，所以無法對於舊元件採取重用。

## 第二節 系統高階規劃階段

### 一、故事建置

產品負責人依據產品待辦清單，產生描述系統主流程的活動圖以及故事描述表。

#### 1. 系統主流程活動圖：

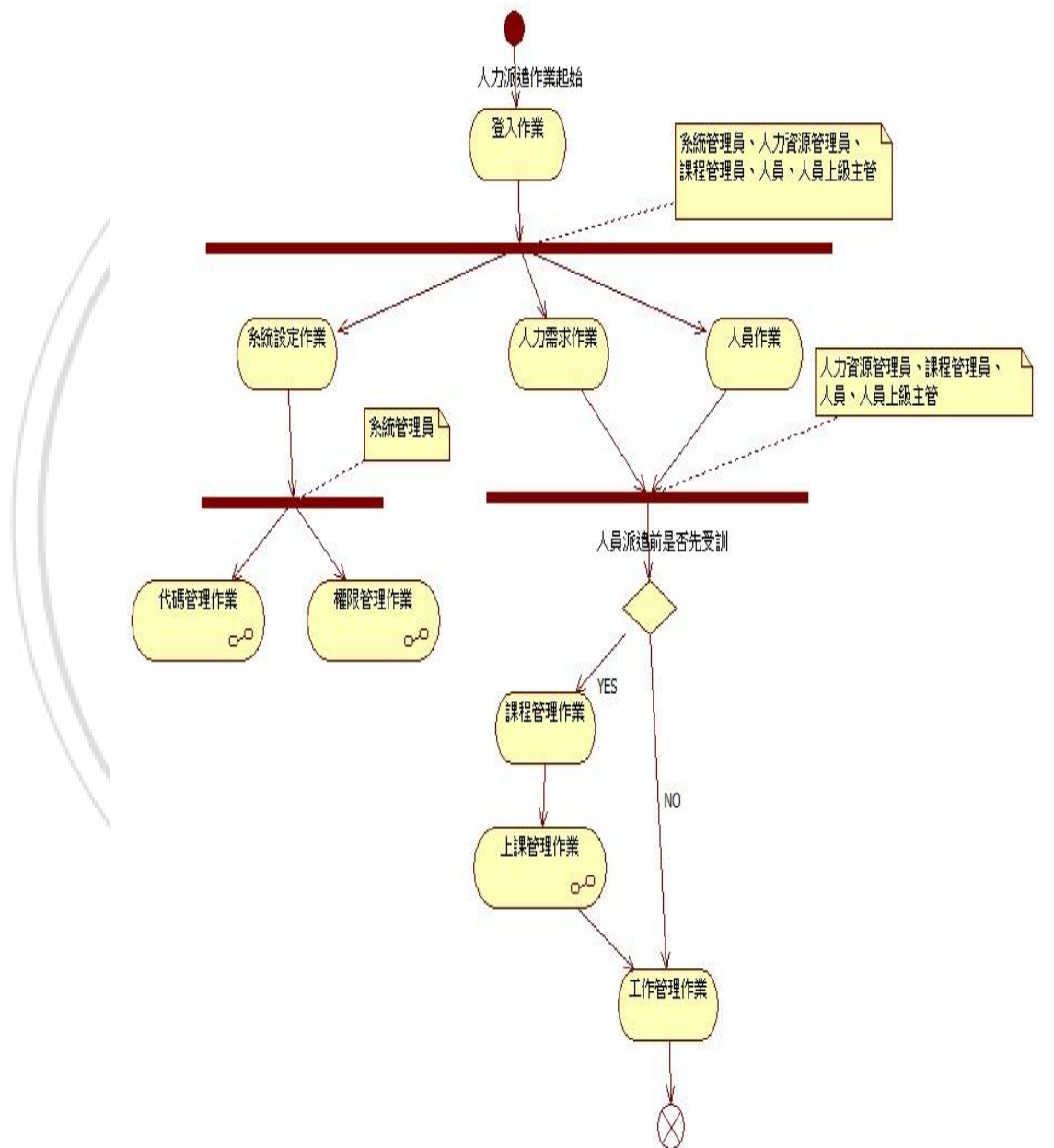


圖 4-4 系統主流程活動圖

資料來源：本研究整理

## 2. 故事描述表：

表 4-1 故事描述表

ID	Name	Description
1	登入	登入畫面包含讓使用者輸入使用者帳號及密碼，並且有驗證處理。
2	新增員工資料	登入之後，擁有人力資源管理員權限者可進行員工資料新增，以 Ajax 方式達成。

資料來源：本研究整理

## 二、框架導入

### 1. Struts 框架

本系統導入 Struts2 框架以快速架構出 MVC 架構分層，Struts2 框架在使用時，是以 WebWork 為核心，幾乎每個類別都會去訪問 WebWork 的功能，可以說 Struts2 是以 WebWork 延伸而來，下圖為 Struts2 結構圖：

# Struts

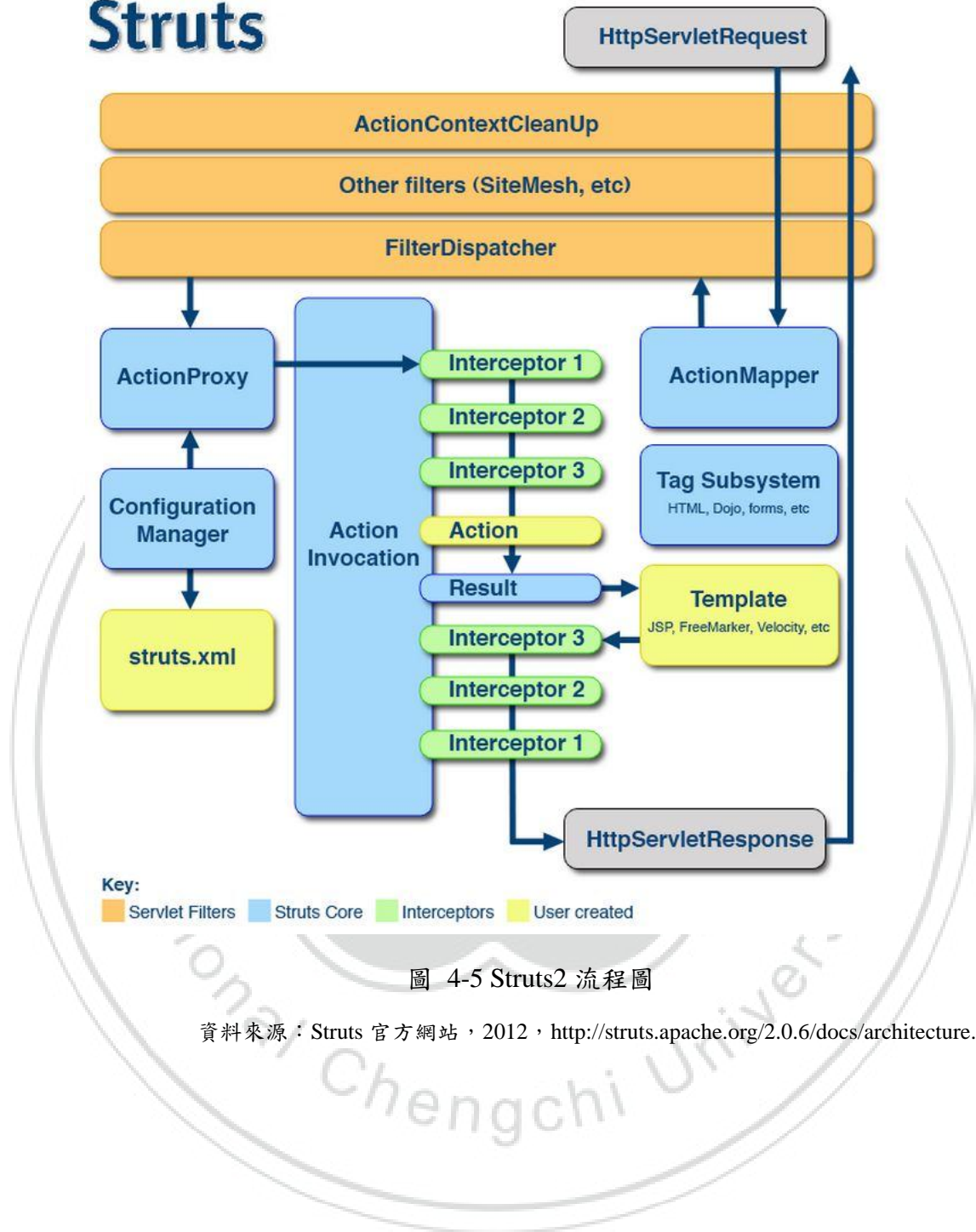


圖 4-5 Struts2 流程圖

資料來源：Struts 官方網站，2012，<http://struts.apache.org/2.0.6/docs/architecture.html>

Struts 生命週期:

(1) 過濾器(Filter)

由使用者送出請求開始，會經由容器的配置檔 web.xml 中的過濾器開始過濾，形成一連串的過濾串，首先是 Struts2 中自帶的 ActionContextCleanUp 過濾器，主要是用來當前線程(Thread)的 ActionContext 以及 Dispatcher 以防止記憶體洩漏問題(Memory Leak)，配置如下圖：

```
<filter>
  <filter-name>struts2-cleanup</filter-name>
  <filter-class>org.apache.struts2.dispatcher.ActionContextCleanUp</filter-class>
</filter>
<filter-mapping>
  <filter-name>struts2-cleanup</filter-name>
  <url-pattern>*/</url-pattern>
</filter-mapping>
```

圖 4-6 Struts2 過濾器

資料來源：本研究整理

接下來是存在於 web.xml 配置檔中其它過濾器進行過濾，例如編碼過濾器，用來將頁面資料編碼成為 UTF-8 編碼，配置如下圖：

```
<filter>
  <filter-name>encodingFilter</filter-name>
  <filter-class>org.springframework.web.filter.CharacterEncodingFilter
</filter-class>
  <init-param>
    <param-name>encoding</param-name>
    <param-value>UTF-8</param-value>
  </init-param>
</filter>
```

圖 4-7 Struts2 過濾器編碼

資料來源：本研究整理

接著則是 Struts2 充當中央控制器的核心過濾器 FilterDispatcher，是 Struts2 框架的基礎，用來處理框架內部的控制流程和處理機制，Struts2 版本 2.0.x 至 2.1.2 是以 FilterDispatcher 為核心過濾器，而在版本 2.1.3 以後則是以 StrutsPrepareAndExecuteFilter 替代。因為過去若將自定義的過濾器放置到 Struts2 過濾器之後，則會如同虛設，但 StrutsPrepareAndExecuteFilter

可以拆分成 StrutsPrepareFilter 和 StrutsExecuteFilter 如此一來就可以在兩者間建立自己所定義的過濾器，配置如下圖：

```
<filter>
  <filter-name>struts2</filter-name>
  <filter-class>org.apache.struts2.dispatcher.ng.filter.StrutsPrepareAndExecuteFilter
</filter-class>
  <init-param>
    <param-name>config</param-name>
    <param-value>
      struts-default.xml,
      struts-plugin.xml,
      edu/nccu/hr/common/struts2/struts.xml,
    </param-value>
  </init-param>
</filter>
<filter-mapping>
  <filter-name>struts2</filter-name>
  <url-pattern>/*</url-pattern>
</filter-mapping>
```

圖 4-8 Struts2 StrutsPrepareAndExecuteFilter

資料來源：本研究整理

## (2) Action 映射器(ActionMapper)

FilterDispatcher 會呼叫 Action 映射器用來判斷請求是否需要 Struts2 處理，以請求參數的後綴字決定，通常是\*.action，也可自行定義後綴字，如果 ActionMapper 決定調用某個 Action，則透過 FilterDispatcher 將請求交給 ActionProxy 處理。

## (3) Action 代理器(ActionProxy)

假如 Action 映射器發現需要請求需要 Struts2 處理，則 FilterDispatcher 將控制權委任給 Action 代理器，由 Action 代理器讀取 Struts2 配置檔 struts.xml 找尋真正執行動作的 Action。

## (4) 攔截器(Interceptor)

在進入到真正執行的 Action 前，會先經過一連串的攔截器，對請求做處理，例如例外處理，通常配置於 struts.xml 中，配置如下圖：

```

<interceptors>
  <interceptor name="LogincheckInterceptor" class="edu.nccu.hr.common.interceptor.LogincheckHandler"></interceptor>
  <interceptor
    class="edu.nccu.hr.common.interceptor.ExceptionHandler"
    name="exceptionInterceptor"/>
  <interceptor
    class="edu.nccu.hr.common.interceptor.JsonAjaxExceptionHandler"
    name="jsonAjaxExceptionInterceptor"/>
  <interceptor
    class="edu.nccu.hr.common.interceptor.SimpleAjaxExceptionHandler"
    name="simpleAjaxExceptionHandler"/>

```

圖 4-9 Struts2 攔截器

資料來源：本研究整理

所經過的攔截器會以堆疊的方式存放形成攔截器串，配置如下圖：

```

<!--logincheck interceptor-->
<interceptor-stack name="LogincheckStack">
  <interceptor-ref name="defaultStack"/>
  <interceptor-ref name="LogincheckInterceptor"/>
</interceptor-stack>

<!--exception interceptor-->
<interceptor-stack name="exceptionStack">
  <interceptor-ref name="defaultStack"/>
  <interceptor-ref name="exceptionInterceptor"/>
</interceptor-stack>

<!-- Ajax return json exception interceptor-->
<interceptor-stack name="jsonAjaxStack">
  <interceptor-ref name="defaultStack"/>
  <interceptor-ref name="jsonAjaxExceptionHandler"/>
</interceptor-stack>

<!-- Ajax return original data type exception interceptor -->
<interceptor-stack name="simpleAjaxStack">
  <interceptor-ref name="defaultStack"/>
  <interceptor-ref name="simpleAjaxExceptionHandler"/>
</interceptor-stack>

<!-- chain interceptor -->
<interceptor-stack name="chainStack">
  <interceptor-ref name="chain"/>
  <interceptor-ref name="basicStack"/>
  <interceptor-ref name="exceptionInterceptor"/>
</interceptor-stack>

```

圖 4-10 Struts2 攔截器串

資料來源：本研究整理

接下來進入真正的 Action 做處理，Action 處理完後，依照 Action 所返回的 Result 會去找對應相關的視圖，例如像是 JSP 頁面，此時若是有用到 Struts2 所提供的標籤庫(tag library)則會先呼叫標籤子系統(Tag Subsystem)。

由於使用框架所提供的標籤，會使得整個專案對於框架相依性大幅提升，未來若框架版本升級或改版，專案都需要大幅修改，例如 Struts2 版本在 2.1.6 以前所提供的 Ajax 標籤是和 Ajax 框架 Dojo 做綁定，於版本 2.1.6 以後，由於支持 Dojo 框架文件過少，Struts2 決定把它單獨分出來，造成 Ajax 標籤用法大幅改變。本研究為了防止這情形出現，所以以 JSP2.0 的標準標籤庫 JSTL(JSP Standard Tag Library)和 EL(Expression Language)來開發系統，如下圖中(c:set 與 c:out 為 JSTL 標籤，\${}則代表 EL):

```
<div title="使用者名稱" selected="true">
  <c:set var="a" scope="session" value="session_login"/>
  <h6><c:out value="${sessionScope.session_login.loginCode}"/></h6>
</div>
```

圖 4-11 JSTL 使用方式

資料來源：本研究整理

呼叫視圖結束後，攔截器由於存放於堆疊中，會以反向順序執行並將回應返回給使用者。



## 2. Spring 框架

Spring 框架主要用於將 Struts 中的 Action 進行實例化以及將依賴注入的工作交給 Spring 容器統一管理。本研究採用的版本為 Spring2.0.8，依據 Spring 官網所提供的技術文件(The Spring Framework - Reference Documentation 2012)整理出 Spring2.0.8 框架與過去版本相比做了以下的改進：

### (1) bean 作用域

bean 的作用域目的是為了定義 bean 在何時對應實例化，如下表所示：

表 4-2 bean 作用域分類表

作用域	描述
Singleton	在每個 Spring IoC 容器中一個 bean 對應一個物件實例
Prototype	一個 bean 定義對應多個物件實例
Request	在一次 HTTP 請求下，一個 bean 對應一個實例。即每次 HTTP 請求將會有各自的 bean 實例，它們依據某個 bean 定義創建而成。該作用域僅在基於 web 的 Spring ApplicationContext 情形下有效。
Session	在一次 HTTP Session 下，一個 bean 對應一個實例。該作用域僅在基於 web 的 Spring Application Context 情形下有效。
Global session	在一個全域的 HTTP Session 中，一個 bean 定義對應一個實例。典型情況下，僅在使用 portlet context 的時候有效。該作用域僅在基於 web 的 Spring ApplicationContext 情形下有效。

資料來源：本研究彙整

配置如下圖：

```
<bean id="exampleInfoActionBean" class="edu.nccu.hr.action.example.ExampleInfoAction" scope="prototype">  
  <property name="exampleInfoService" ref="exampleInfoServiceBean"/>  
</bean>
```

圖 4-12 bean 作用域配置

資料來源：本研究整理

表示這個 Action bean 可對應多個物件實例。

## (2) 簡單的 XML 配置

新的基於 XML Schema 的 XML 配置語法產生，Spring 的 XML 配置變得  
更簡單。原有的配置如下圖：

```
<beans xmlns="http://www.springframework.org/schema/beans"  
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
  xsi:schemaLocation="http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans.xsd">
```

圖 4-13 原 Spring XML 配置

資料來源：本研究整理

現在可將此配置簡化如下圖：

```
<!DOCTYPE beans PUBLIC "-//SPRING//DTD BEAN 2.0//EN"  
  "http://www.springframework.org/dtd/spring-beans-2.0.dtd"  
<beans>
```

圖 4-14 新 Spring XML 配置

資料來源：本研究整理

## (3) Annotation(註釋)驅動配置

Spring2.0 引入了一些用於配置的 annotation，要使用註釋標籤必須使用  
Java5 或以上的版本，註釋標籤如下表所整理：

表 4-3 Spring2.0 註釋標籤

註釋標籤	描述
@Autowired	自動讓 Spring 通過檢查 BeanFactory 中的內容，來指定所配置的 bean。
@Resource	功能與 Autowired 類似，可通過 name 屬性進行指定所配置的 bean，此標籤屬於 J2EE 使用此標籤可以降低與 Spring 框架的耦合。
@PostConstruct	在其生命週期與 Spring 生命週期介面的方法或者顯示宣告回調方法同一刻上調用。
@PreDestroy	

資料來源：本研究彙整

### 3. Hibernate 框架

Hibernate 框架主要用於 ORM 的實現，實現步驟如下：

#### (1) Hibernate 配置文件(\*.cfg.xml)

配置 JDBC 驅動類別全名、JDBC 所連接的 URL、使用者帳戶及密碼、配置資料庫方言(例如 MySQL 為 MySQLDialect) 以及註冊 ORM 映射文件。

```
<!DOCTYPE hibernate-configuration PUBLIC
    "-//Hibernate/Hibernate Configuration DTD 3.0//EN"
    "http://hibernate.sourceforge.net/hibernate-configuration-3.0.dtd">
<hibernate-configuration>
  <session-factory>
    <property name="hibernate.connection.driver_class">com.mysql.jdbc.Driver</property>
    <property name="hibernate.connection.password">****</property>
    <property name="hibernate.connection.url">jdbc:mysql://localhost:3306/odwebdb</property>
    <property name="hibernate.connection.username">root</property>
    <property name="hibernate.dialect">org.hibernate.dialect.MySQLDialect</property>
  </session-factory>
</hibernate-configuration>
```

圖 4-15 Hibernate 配置文件

資料來源：本研究整理

本研究的註冊 ORM 映射文件部分，在整合框架的時候，轉交給 Spring 框架做處理，故在此配置中不會看到相關處理。

#### (2) ORM 映射文件(\*.xml)：

將 Java 中的實體對應到資料庫中的資料表，配置如下圖：

```
<!DOCTYPE hibernate-mapping PUBLIC "-//Hibernate/Hibernate Mapping DTD 3.0//EN"
    "http://hibernate.sourceforge.net/hibernate-mapping-3.0.dtd">
<!-- Generated 2012/4/24 ?U?? 03:53:29 by Hibernate Tools 3.4.0.CR1 -->
<hibernate-mapping package="edu.nccu.hr.model.example">
  <class name="ExampleInfo" table="employee"> <!-- class name == table name -->
    <id name="employee_id" type="java.lang.Integer">
      <column name="employee_id" />
      <generator class="identity" />
    </id>
    <property name="employee_name" type="java.lang.String" not-null="false" length="100">
      <column name="employee_name" />
    </property>
    <property name="skillset_id" type="java.lang.Integer" not-null="false">
      <column name="skillset_id" />
    </property>
    <property name="dept_id" type="java.lang.Integer" not-null="false">
      <column name="dept_id" />
    </property>
  </class>
</hibernate-mapping>
```

圖 4-16 Hibernate 映射文件

資料來源：本研究整理

#### 4. JQuery 框架

JQuery 框架是一個 JavaScript 框架，透過函式庫簡化了 JavaScript，可以讓使用者更方便的處理 HTML 文件檔、events 事件以及 Ajax 互動，也是目前最廣為使用的框架。

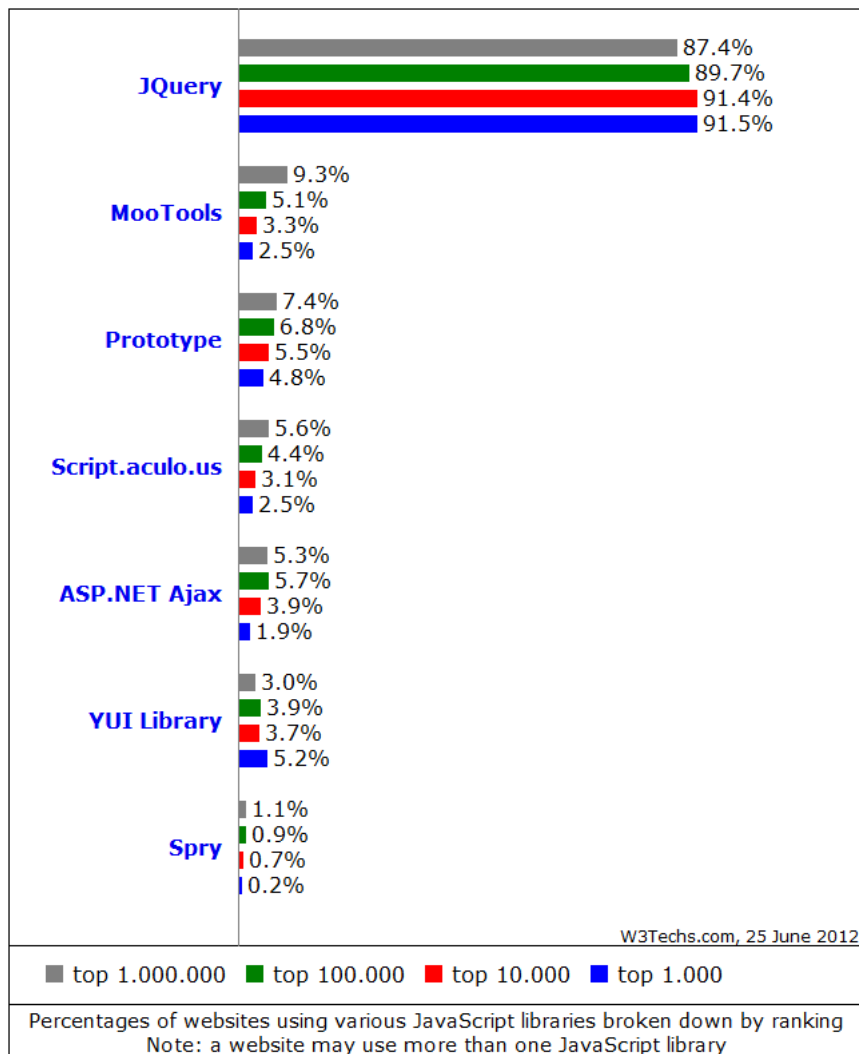


圖 4-17 JavaScript 框架使用比例

資料來源：World Wide Web Technology Surveys，2012，

[http://w3techs.com/technologies/cross/javascript\\_library/ranking](http://w3techs.com/technologies/cross/javascript_library/ranking)

故本研究採以 JQuery 框架來達到系統之 AJAX 功能，利用前端 JavaScript 去呼叫 Struts 框架，再利用後端三個框架的整合做資料處理，使前端使用者在等待數據傳回時，仍可對當前頁面其它部分進行操作，提升使用者體驗。

JQuery 框架透過 \$ 當作選擇器，簡化了過去 JavaScript 必須透過 DOM 來擷取節

點的冗長程式碼，如下圖所示：

```
function clearCondition(){
    $('#queryForm').form('clear');
    query();
}
```

圖 4-18 JQuery 選擇器使用方式

資料來源：本研究整理

使用 JQuery 框架可使 JavaScript 程式碼變得相當簡潔，JQuery 也提供 AJAX 的應用介面(API)提升開發者開發 AJAX 的速度。

## 5. 框架整合

### (1) Spring 框架與 Struts 框架整合

將 struts.xml 裡的 ObjectFactory 設置為 spring：

```
<constant name="struts.objectFactory" value="spring"></constant>
```

圖 4-19 Spring 與 Struts 整合

資料來源：本研究整理

再把 Struts 所配置的 Action，配置到 Spring 的配置文件中，由 Spring 來為 Action 實例化：

```
<bean id="exampleInfoActionBean" class="edu.nccu.hr.action.example.ExampleInfoAction" scope="prototype">
    <property name="exampleInfoService" ref="exampleInfoServiceBean"/>
</bean>
```

圖 4-20 Spring 實例化 Struts Action

資料來源：本研究整理

這樣就可以透過 Spring 來管理 Action 並透過 Spring 來進行依賴注入。

### (2) Spring 框架與 Hibernate 框架整合：

將 Hibernate 的 DataSource、SessionFactory 以及 Hibernate Transaction Manager 交給 Spring 管理。

資料庫配置：

```
<!-- 配置DBCP資料庫 -->
<bean id="dbcp_dataSource" class="org.apache.commons.dbcp.BasicDataSource"
    destroy-method="close">
    <property name="driverClassName">
        <value>${jdbc.driverClassName}</value>
    </property>
    <property name="url">
        <value>${jdbc.url}</value>
    </property>
    <property name="username">
        <value>${jdbc.username}</value>
    </property>
    <property name="password">
        <value>${jdbc.password}</value>
    </property>

    <property name="defaultAutoCommit">
        <value>${jdbc.defaultAutoCommit}</value>
    </property>
</bean>
```

圖 4-21 Spring 管理 Hibernate 資料庫配置

資料來源：本研究整理

SessionFactory 配置：

```
<bean id="sessionFactory"
    class="org.springframework.orm.hibernate3.LocalSessionFactoryBean">
    <property name="dataSource" ref="dbcp_dataSource" />
    <!--ORM映射文件:mappingResources-->
    <!-- ORM目錄 -->
    <property name="mappingDirectoryLocations">
        <value>classpath:edu/nccu/hr/model/hbm/</value>
    </property>

    <property name="hibernateProperties">
        <props>
            <!-- 資料庫語言 -->
            <prop key="hibernate.dialect">${jdbc.hibernate.dialect}</prop>
            <!-- 控制台是否列印SQL -->
            <prop key="hibernate.show_sql">${jdbc.hibernate.show_sql}</prop>
            <!-- 控制台是否格式化SQL語句顯示樣式 -->
            <prop key="hibernate.format_sql">${jdbc.hibernate.format_sql}</prop>
            <!-- 啟動伺服器時，檢查hbm是否與資料庫對應，不對應拋出異常 -->
            <prop key="hibernate.hbm2ddl.auto">validate</prop>

        </props>
    </property>
</bean>
```

圖 4-22 Spring 管理 Hibernate SessionFactory

資料來源：本研究整理

TransactionFactory 配置：

```
<!-- 配置交易(session)管理 -->
<bean id="transactionManager"
      class="org.springframework.orm.hibernate3.HibernateTransactionManager">
  <property name="sessionFactory">
    <ref local="sessionFactory" />
  </property>
</bean>
```

圖 4-23 Spring 管理 Hibernate TransactionFactory

資料來源：本研究整理

### (3) JQuery 與 SSH 框架整合

透過 JavaScript 發出請求，藉由請求參數的後綴字交由 Struts 的過濾器將此請求過濾做處理。

```
function zTreeOnClick(event, treeId, treeNode) {
  addTab("tabId_"+treeNode.ename,treeNode.name,"/HRSystem_SSH2/hr/example/goexampleInfoMain.action");
}
```

圖 4-24 JQuery 與 SSH 整合方式

資料來源：本研究整理

### 三、資料庫分析設計

由 Scrum Master 依據產品負責人所建出之活動圖以及類別圖，對整體系統進行資料庫規劃與設計產生 ERD 圖。

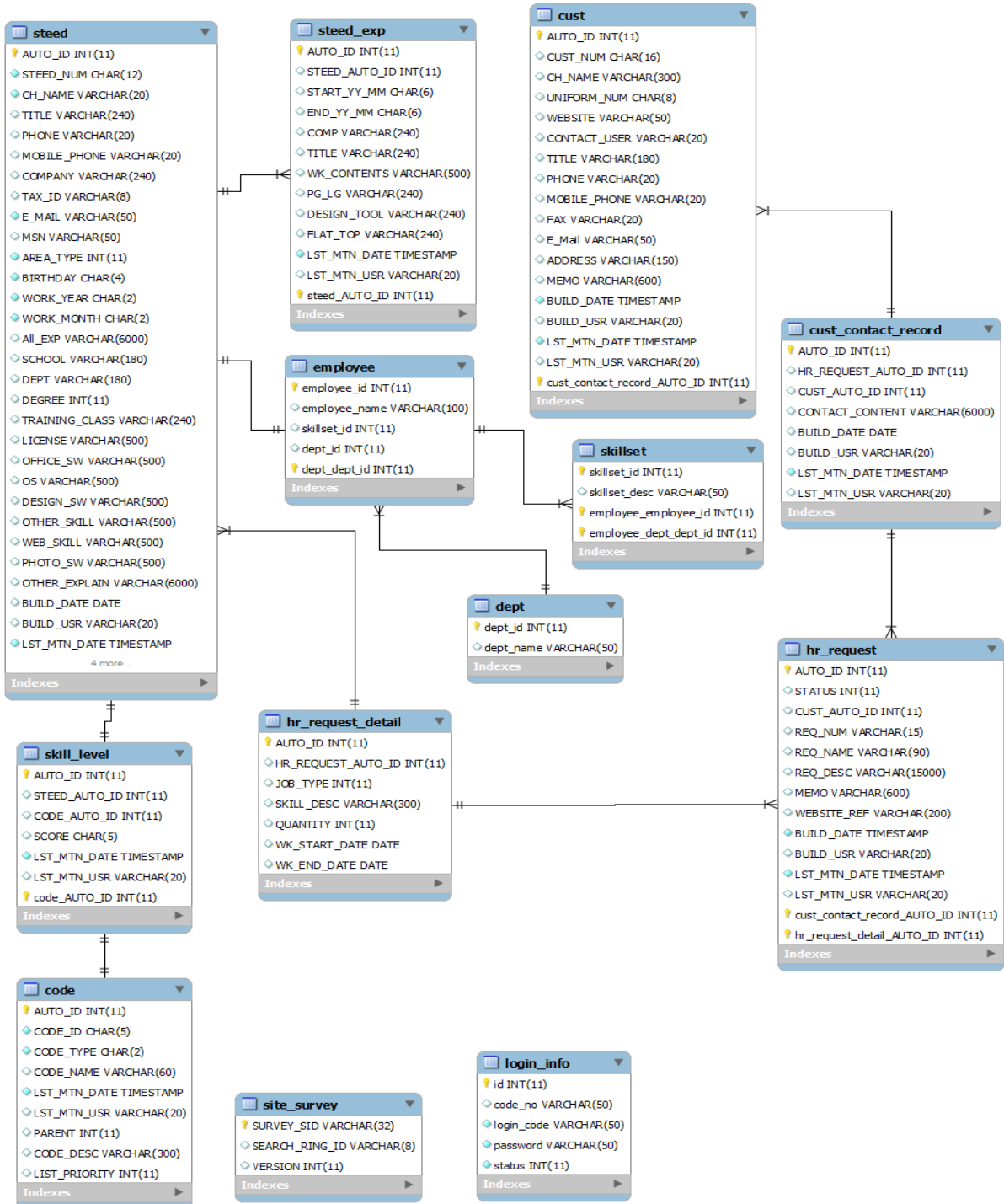


圖 4-25 系統資料庫 ERD 圖



對於資料庫欄位可產生相對應的欄位說明表單，使開發人員可快速了解各欄位所代表之意義：(以 CUST-客戶表格為例)

表 4-4 資料庫欄位說明表

欄位代號	資料型態	欄位名稱	說明
AUTO_ID	INT(11)	自動序號	主鍵
CUST_NUM	CHAR(16)	客戶編號	
CH_NAME	VARCHAR(300)	客戶名稱(中)	
UNIFORM_NUM	CHAR(8)	統編	
WEBSITE	VARCHAR(50)	公司網站	
CONTACT_USER	VARCHAR(20)	公司聯絡人	
TITLE	VARCHAR(180)	聯絡人職稱	
PHONE	VARCHAR(20)	電話	
MOBILE_PHONE	VARCHAR(20)	手機	
FAX	VARCHAR(20)	傳真	
E_MAIL	VARCHAR(50)	電子郵件	
ADDRESS	VARCHAR(150)	地址	
MEMO	VARCHAR(600)	附註	
BUILD_DATE	TIMESTAMP	建檔日期	
BUILD_USR	VARCHAR(20)	建檔人員	
LST_MTN_DATE	TIMESTAMP	最後維護日期	
LST_MTN_USR	VARCHAR(20)	最後維護人員	

在設計出資料庫各實體關聯關係後，進行類別圖的繪製，將程式碼中所會宣告的屬性與方法成員以圖示方式表達：

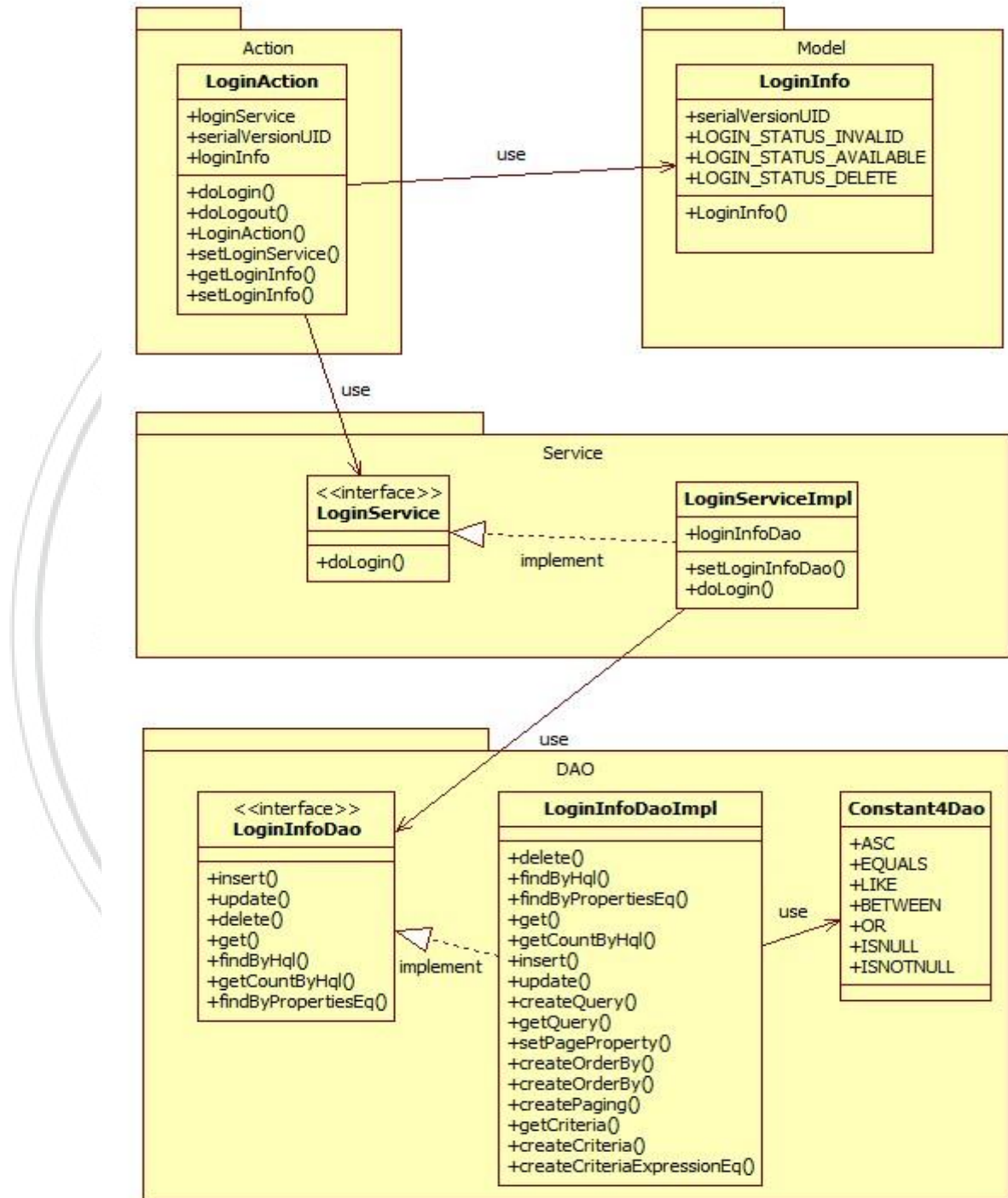


圖 4-26 系統登入功能類別圖

資料來源：本研究整理

#### 四、實作指引文件

##### 1. 基礎設施

用於描述系統概觀的基礎設施。

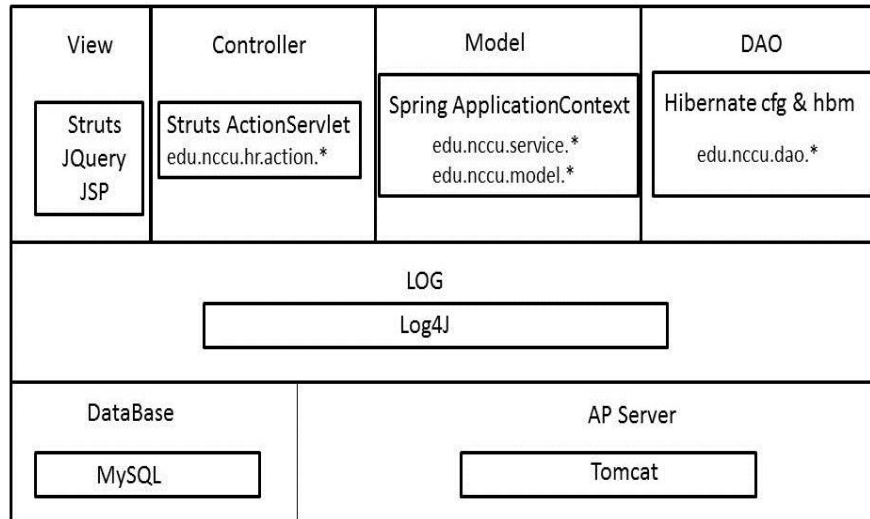


圖 4-27 系統基礎設施

資料來源：本研究整理

由上圖可看出各分層所使用之框架，所選擇的資料庫，以及執行應用程式的伺服器。並製作 Package 分類表，使團隊成員了解 Package 命名規則，以及各 Package 之功用。

表 4-5 系統 package 分類表

Package	說明
Src	Source 相關
<b>HRSystem_SSH2.edu.nccu.hr.common.*</b>	將共用部分抽象出來當元件以利其他專案重複使用。
<b>HRSystem_SSH2.edu.nccu.hr.action.*</b>	Action
<b>HRSystem_SSH2.edu.nccu.service.*</b>	Service
<b>HRSystem_SSH2.edu.nccu.dao.*</b>	DAO
<b>HRSystem_SSH2.edu.nccu.dto.*</b>	DTO
<b>HRSystem_SSH2.edu.nccu.model.*</b>	POJO
<b>HRSystem_SSH2.edu.nccu.constatnt.*</b>	系統常用相關常數

Config	配置檔相關
<b>HRSYSTEM_SSH2.edu.nccu.common.*</b>	共用配置檔(ApplicationContext.xml & struts.xml & interceptor-common-exception.xml)
<b>HRSYSTEM_SSH2.edu.nccu.example.*</b>	個別 struts2 與 spring 配置檔
<b>HRSYSTEM_SSH2.edu.nccu.login.*</b>	個別 struts2 與 spring 配置檔
<b>HRSYSTEM_SSH2.edu.nccu.model.*</b>	Hibernate 相關配置檔(hibernate.cfg.xml & model.hbm.xml)
<b>log4j.properties</b>	Log 相關配置檔
<b>mysql-jdbc.properties</b>	Mysql 相關配置檔

資料來源：本研究整理

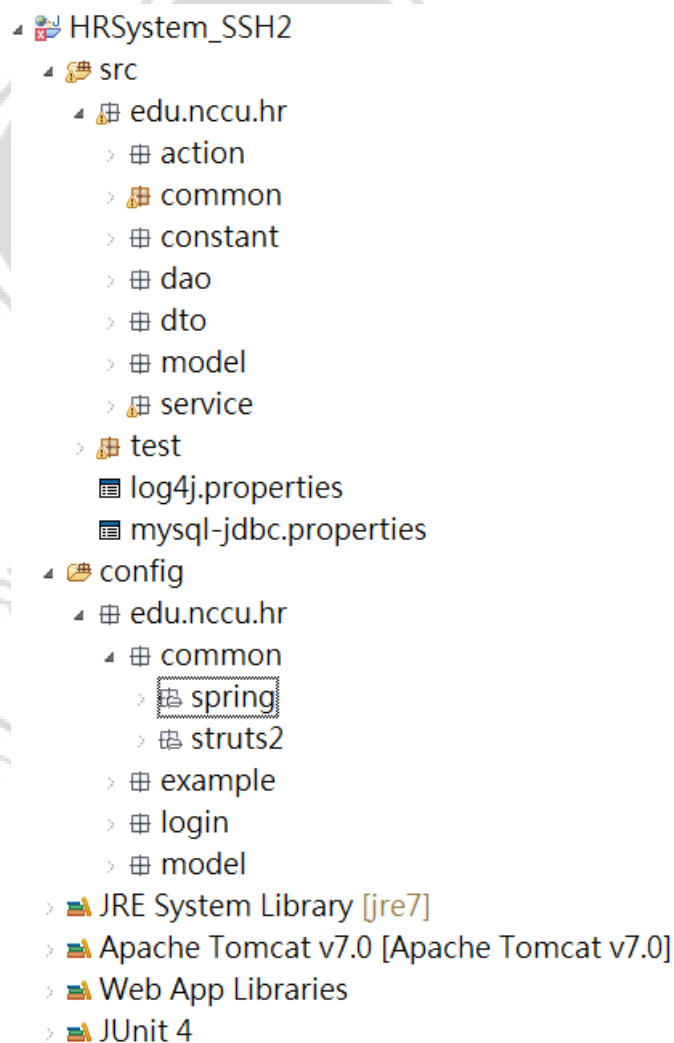


圖 4-28 系統檔案結構圖

資料來源：本研究整理

## 2. 整體架構循序圖

可使團隊成員了解整體程式架構分層流程走向，清楚地顯示出如何分層，統一分層架構。

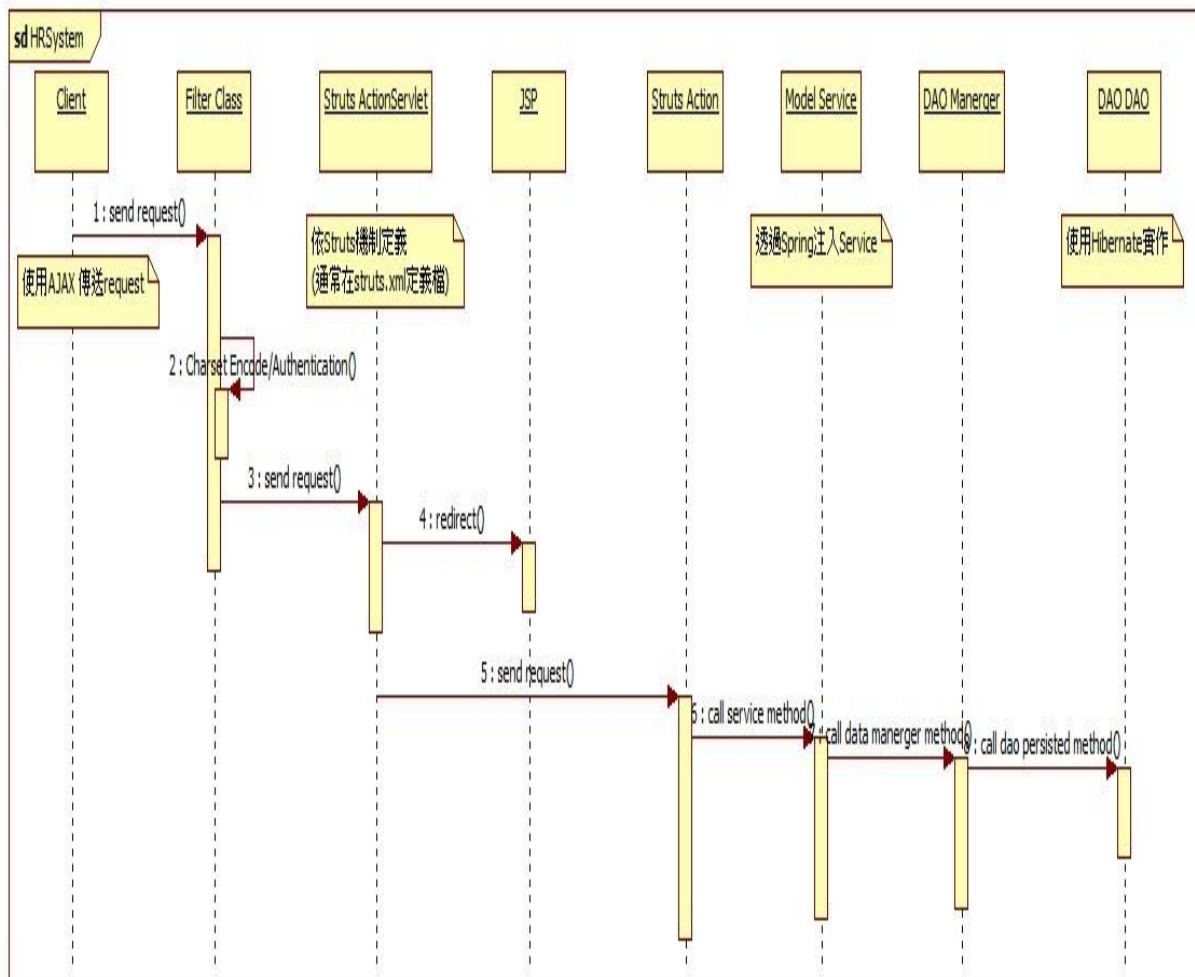


圖 4-29 系統分層循序圖

資料來源：本研究整理

### 3. 開發平台

記錄開發文件所存放位置以及程式碼目錄結構。

- (1) HRSystem 文件存放位置: \\140.119.19.188\研發專案\人力資源管理系統  
     \2012\Guide
- (2) Source 目錄結構

表 4-6 系統 Source 目錄結構

\WebContent	Index.jsp
\WebContent\Web\ common\	共用 css、圖片、javascript、JSP 頁面
\WebContent\Web\ example\	個別 css、圖片、javascript、JSP 頁面
\WebContent\Web\ login\	個別 css、圖片、javascript、JSP 頁面
\WebContent\Web\report\	報表
\ WebContent \WEB-INF\	相關設定檔
\ WebContent \WEB-INF\lib\	函式庫
\ WebContent \WEB-INF\tlds\	標籤
\src\	程式原始碼
\config\	配置檔

資料來源：本研究整理

### 4. 開發環境

統一團隊成員的開發環境，以防出現因環境不同所造成之問題。

表 4-7 系統開發環境

DataBase	<p>MySQL 5/ <b>connector 5.1.10</b></p> <ul style="list-style-type: none"> <li>• Test server Ip:port: localhost:3306</li> <li>• Production server ip: 140.119.19.188/phpMyAdmin/</li> <li>• UserName: xxxx</li> <li>• Password: xxxx</li> </ul>
----------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

AP Server	Tomcat <b>7.0</b>  • Test server IP: localhost:8080  • Production server IP:140.119.19.188:8080
JRE	<b>JRE 1.7.0_03</b>
開發環境	DB Name: odwebdb  Web Application Name: HRSystem_SSH2
正式環境	DB Name: odwebdb  Web Application Name: HRSystem_SSH2
IDE	<b>Eclipse 3.7(indigo)</b>

資料來源：本研究整理

#### 5. 開發工具

建議開發團員盡量使用相同之開發工具，以防出現檔案無法讀取或附加元件不同等問題。

- 文件撰寫：Microsoft word 2010
- 程式開發：Eclipse 3.7(indigo)
- 開發框架：Strus2 + Spring2 + Hibernate3 + JQuery

#### 6. 共用函式庫

防止團隊成員因為函式庫版本不同所發生之衝突，或使用不同版本所依賴相關的函式庫不同。

表 4-8 系統共用函式庫

• antlr 2.7.6rc1	• google-collections 1.0
• asm 2.2.3	• hibernate 3.0
• asm-commons 2.2.3	• json-lib 2.1
• asm-util 2.2.3	• jsonplugin 0.32
• spring-aspects 3.0.5.RELEASE	• jstl

• bonecp 0.7.0	• jta
• bonecp-spring 0.7.0	• ognl 2.7.3
• bonecp-provider 0.7.0	• pinyin4j 2.5.0
• cglib-nodep 2.1_3	• spring 2.0.8
• commons-beanutils 1.7.0	• standard
• commons-collections 3.2	• struts2-core 2.0.8.1
• commons-fileupload 1.2.1	• struts2-spring-plugin 2.0.8.1
• commons-lang 2.3	• xwork-core 2.1.6
• commons-logging 1.1.1	• mysql-connector-java 5.1.10
• commons-pool 1.4	• servlet-api
• commons-validator 1.3.1	• commons-dbcp
• dom4j 1.6.1	• el-api 1.0
• ehcache 1.2.3	• jsp-api 2.1
• ezmorph 1.0.3	• log4j 1.2.15
• freemarker 2.3.15	• slf4j-api 1.5.8
	• slf4j-log4j12 1.5.8

資料來源：本研究整理



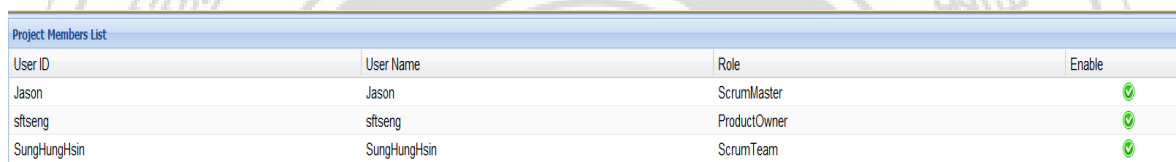
### 第三節 反覆執行階段

一個系統專案的開發，是由許多衝刺所組成，本研究以開發人力資源管理系統雛型為例，藉以演示此階段的操作。而操作工具則是使用由台北科技大學團隊所研發的 ezScrum 軟體來做為 Scrum 執行記錄(資料來源：ezScrum，2011，<http://scrum.tw/index.php/tw/home>)。

#### 一、衝刺規劃

##### 1. 團隊分工

本研究 Scrum 分工團隊成員為 3 人，其中一人擔任產品負責人，一人擔任 Scrum Master，最後一人為程式開發者。



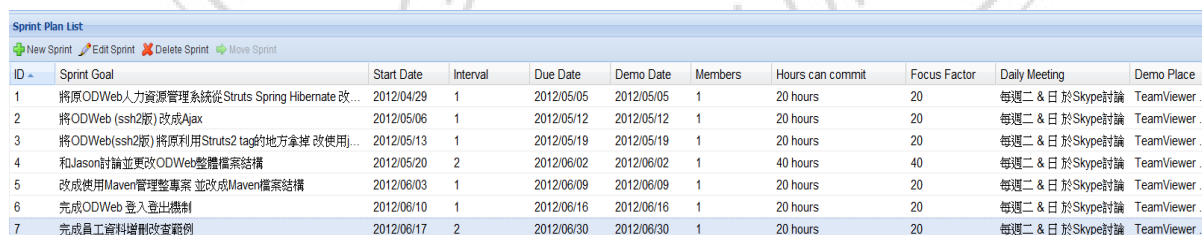
User ID	User Name	Role	Enable
Jason	Jason	ScrumMaster	✓
sftseng	sftseng	ProductOwner	✓
SungHungHsin	SungHungHsin	ScrumTeam	✓

圖 4-30 團隊分工

資料來源：本研究整理

產品負責人負責需求制訂，Scrum Master 負責執行 Scrum 的困難排除以及擔任技術開發問題顧問，團隊成員則進行開發。

##### 2. 衝刺目標與長度決定



ID	Sprint Goal	Start Date	Interval	Due Date	Demo Date	Members	Hours can commit	Focus Factor	Daily Meeting	Demo Place
1	將原ODWeb人力資源管理系統從 Struts Spring Hibernate 改...	2012/04/29	1	2012/05/05	2012/05/05	1	20 hours	20	每週二 & 日 於 Skype 討論	TeamViewer ...
2	將 ODWeb (ssh2版) 改成 Ajax	2012/05/06	1	2012/05/12	2012/05/12	1	20 hours	20	每週二 & 日 於 Skype 討論	TeamViewer ...
3	將 ODWeb (ssh2版) 將原利用 Struts2 tag 的地方拿掉 改使用...	2012/05/13	1	2012/05/19	2012/05/19	1	20 hours	20	每週二 & 日 於 Skype 討論	TeamViewer ...
4	和 Jason 討論並更改 ODWeb 整體檔案結構	2012/05/20	2	2012/06/02	2012/06/02	1	40 hours	40	每週二 & 日 於 Skype 討論	TeamViewer ...
5	改成使用 Maven 管理專案 並改成 Maven 檔案結構	2012/06/03	1	2012/06/09	2012/06/09	1	20 hours	20	每週二 & 日 於 Skype 討論	TeamViewer ...
6	完成 ODWeb 登入登出機制	2012/06/10	1	2012/06/16	2012/06/16	1	20 hours	20	每週二 & 日 於 Skype 討論	TeamViewer ...
7	完成員工資料增刪改查範例	2012/06/17	2	2012/06/30	2012/06/30	1	20 hours	20	每週二 & 日 於 Skype 討論	TeamViewer ...

圖 4-31 衝刺目標與長度

資料來源：本研究整理

由上圖可看出每一次衝刺所決定的目標以及衝刺長度，衝刺的目標是依據和產品負責人討論該其衝刺所應該要執行的故事，再和 Scrum Master 討論可行性及評估

時間。此專案已經經過 7 次的衝刺，花費 9 個禮拜的時間，所達成的衝刺目標分別為：

- 將原 ODWeb 人力資源管理系統從 Struts Spring Hibernate 改版成 Struts2 Spring2 Hibernate3(為期一週)
- 將 ODWeb (ssh2 版) 改成 Ajax(為期一週)
- 將 ODWeb(ssh2 版) 將原利用 Struts2 tag 的地方拿掉 改使用 jstl tag 並將原本使用的 Ajax 框架由 Dojo 改成 JQuery(為期一週)
- 和 Scrum Master 討論並更改 ODWeb 整體檔案結構(為期兩週)
- 改成使用 Maven 管理整專案 並改成 Maven 檔案結構(為期一週)
- 完成 ODWeb 登入登出機制(為期一週)
- 完成員工資料增刪改查範例(為期兩週)

## 二、技術故事

本研究所進行之 7 項衝刺目標，前 5 項屬於技術故事，是因為 SSH 版本不同，使用方式與舊版本差異極大，再加上將使用者體驗考慮進去，所以使用了 AJAX 技術，另外為了減少對特定框架的相依性，將原開發版本的 Struts 標籤拿掉，改成 JSP 通用的 JSTL 標籤取代。

由於開發的是系統離型，未來可供其他團隊繼續進行開發，故在檔案結構上更動了許多次，雖然說每種結構都有其優缺點，但還是與有 SSH 開發經驗的 Scrum Master 進行一系列的討論，以達到清晰易懂的檔案結構，以利未來的開發。

屬於技術故事的項目如下：

- 將原 ODWeb 人力資源管理系統從 Struts Spring Hibernate 改版成 Struts2 Spring2 Hibernate3(為期一周)
- 將 ODWeb (ssh2 版) 改成 Ajax(為期一周)
- 將 ODWeb(ssh2 版) 將原利用 Struts2 tag 的地方拿掉 改使用 jstl tag 並將原本

使用的 Ajax 框架由 Dojo 改成 JQuery(為期一周)

- 和 Scrum Master 討論並更改 ODWeb 整體檔案結構(為期兩周)
- 使用 SVN 做版本管理加上使用 Maven 管理專案 並改成 Maven 檔案結構(為期一周)

本專案用 Maven 來對專案整體做管理，利用 pom.xml 管理專案函式庫的依賴關係。

Maven 專案目錄結構：

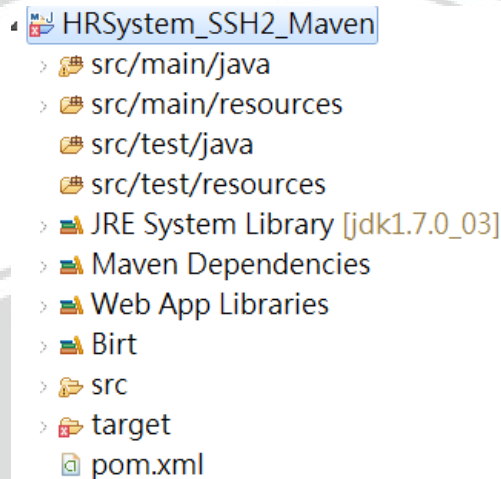


圖 4-32 Maven 目錄結構

資料來源：本研究整理

- src/main/java：放置 java 類別檔的目錄。
- src/main/resources：放置資源相關配置檔。
- src/test/java：放置 JUnit 所進行單元測試的類別檔。
- src/test/resources：放置測試所需之資源配置檔。

經過以上這些技術故事，使得整體專案的一致性獲得提升，自動性的管理也減少了人力管理的浪費以及人力管理可能造成的錯誤。

### 三、衝刺待辦清單

團隊於衝刺開始前，所進行對於此次衝刺所要完成哪些故事的討論，並評估每個故事的重要性以及粗略估計完成所需時間，並對於過於龐大的故事進行任務分解動作，產生衝刺待辦清單。

Release #None; Sprint #7 - 完成員工資料增刪修改紀錄 | Story Point : 12.0; Task Hours : 12.0; Total Hours can commit : 4.0

ID	Tag	Name	Importance	Value	Estimate	Handler	Status	06/25	06/26	06/27	06/28	06/29	07/02	07/03	07/04	07/05	07/06
19		人員基本資料維護	80	80	12		new										
40		刪除員工			3	SungHungHsin	closed	3.0	3.0	0.0							
39		新增員工			3	SungHungHsin	closed	3.0	0.0	0.0							
41		修改員工			3	SungHungHsin	assigned	3.0	3.0	3.0							
44		查詢員工			3	SungHungHsin	assigned	3.0	3.0	3.0							

圖 4-33 衝刺待辦清單

資料來源：本研究整理

#### 四、每日 Scrum 會議

由於成員時間無法全部投入於此專案中，依照 Williams 研究指出(Williams et al. 2011)，Scrum 是可以依據成員人數、環境等因素進行彈性變形，所以每日的 Scrum 會議，改成一星期兩次，這種調度對於大型專案可能很容易會失去控制，但由於本研究成員較少，所以太頻繁的開會反而會造成開發成員進行開發的時間減少，而造成專案進度延遲。

#### 五、任務分解

將龐大的故事再進一步的分解成細部任務，再由團隊成員各自將任務領回去，由任務板記錄每日的工作概況，以及各任務的完成程度。



圖 4-34 任務板

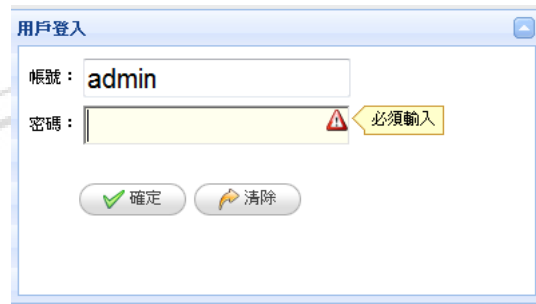
此任務板上方顯示了任務以及故事的燃盡圖，可看出每天的任務以及故事執行進度，下方則顯示未繳交(Not Checked Out)、繳交(Checked Out)以及完成(Done)三大進度項目，未繳交表示尚未提交至版本管理伺服器，繳交則代表已經上傳至版本管理伺服器但尚未通過檢測還未符合完成的定義，而完成則代表通過測試符合完成標準。



## 六、衝刺檢視

向專案之利害關係人展示此回合衝刺檢視的成果，最好的方式為讓客戶操作產出成果。以衝刺七展示成果為例：

### 1. 登入功能



用戶登入

帳號： admin

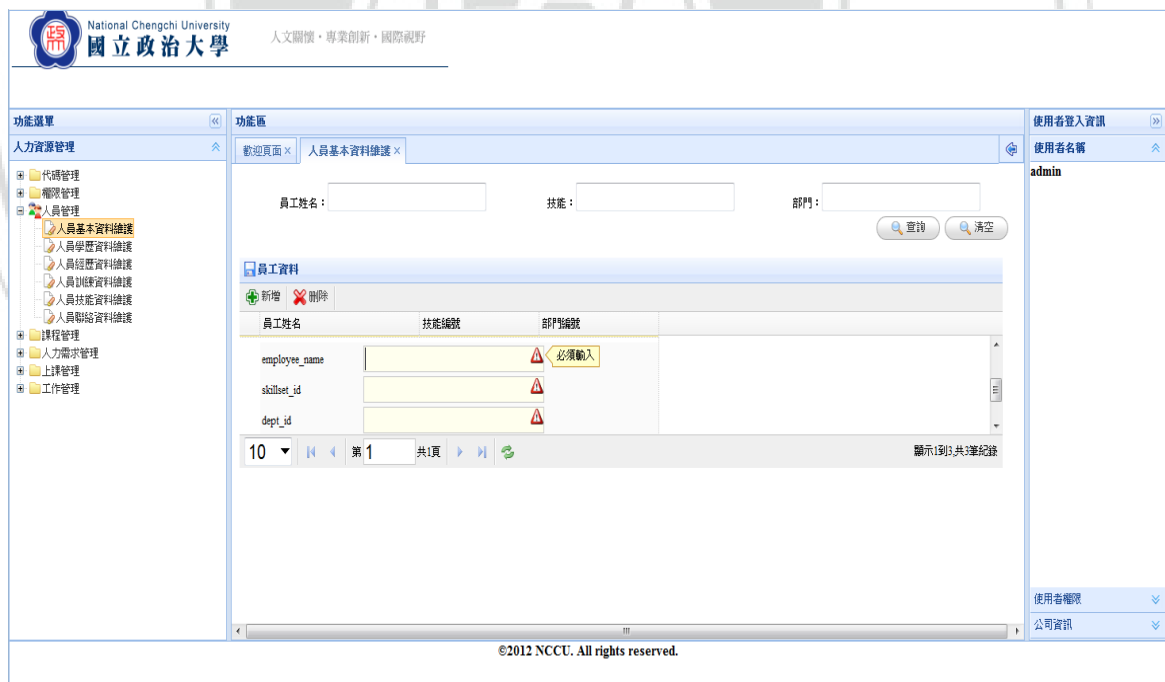
密碼：  必須輸入

確定 清除

圖 4-35 系統登入功能

資料來源：本研究整理

### 2. 新增功能



National Chengchi University  
國立政治大學 人文關懷·專業創新·國際視野

功能選單：人力資源管理 > 人員管理 > 人員基本資料維護

功能區：歡迎頁面 | 人員基本資料維護

使用者登入資訊：admin

員工姓名：  技能：  部門：  查詢 清空

員工資料

新增 刪除

員工姓名	技能編號	部門編號
employee_name	<input type="text"/> <span>必須輸入</span>	
skillsset_id	<input type="text"/> <span>必須輸入</span>	
dept_id	<input type="text"/> <span>必須輸入</span>	

10 第 1 頁 共 1 頁 顯示 1 到 3 筆紀錄

©2012 NCCU. All rights reserved.

圖 4-36 員工資料新增功能

資料來源：本研究整理

### 3. 修改功能

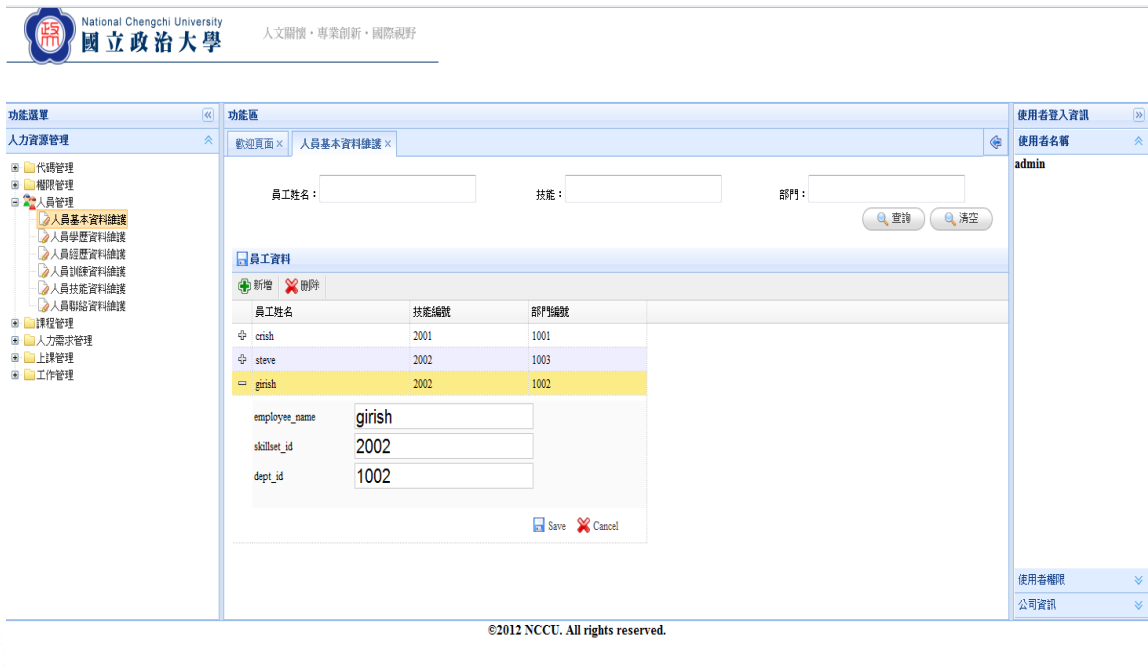


圖 4-37 員工資料修改功能

資料來源：本研究整理

### 4. 刪除功能

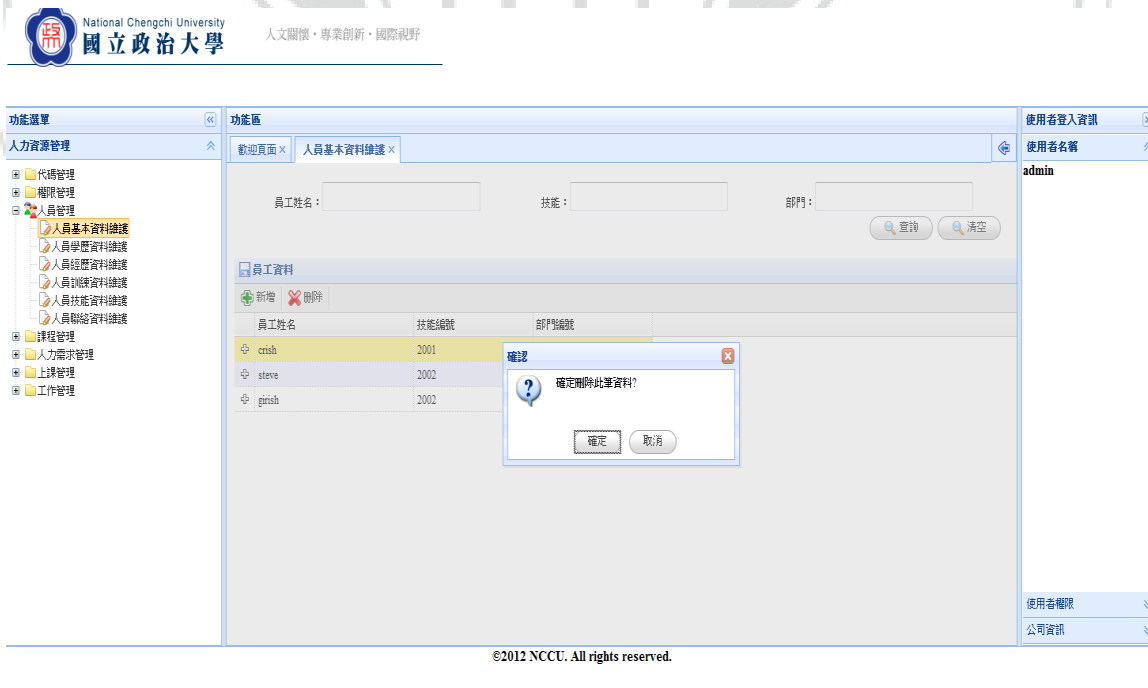


圖 4-38 員工資料刪除功能

資料來源：本研究整理

## 5. 動態條件查詢功能

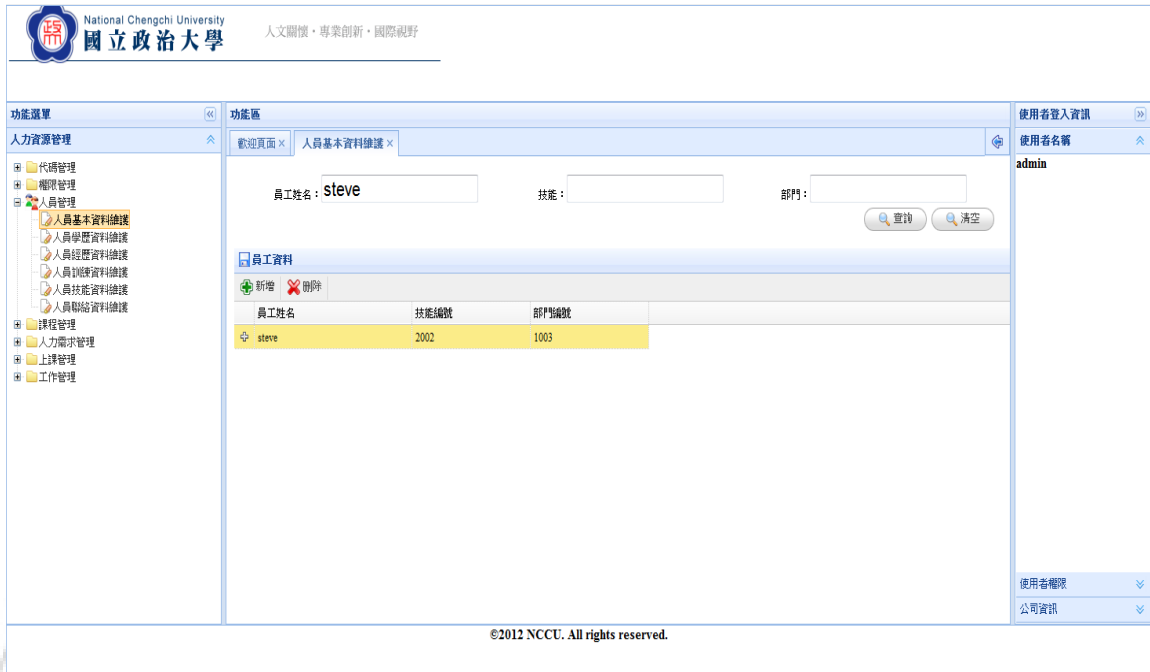


圖 4-39 員工資料動態條件查詢功能

資料來源：本研究整理

## 6. 登出功能

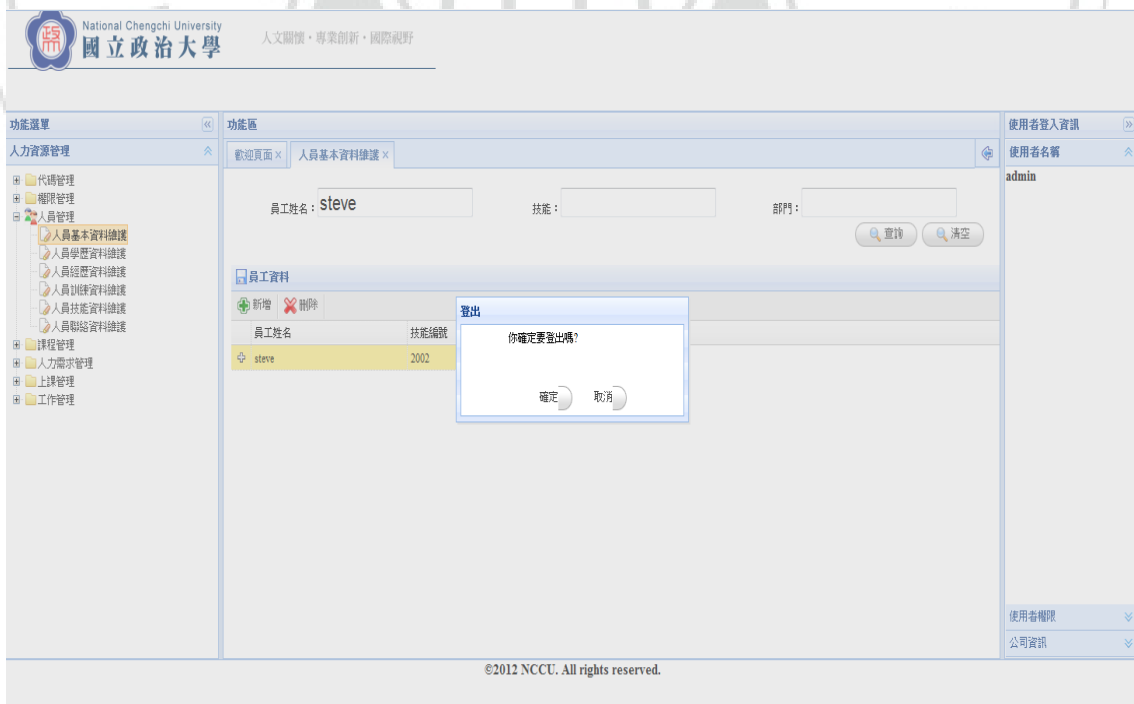


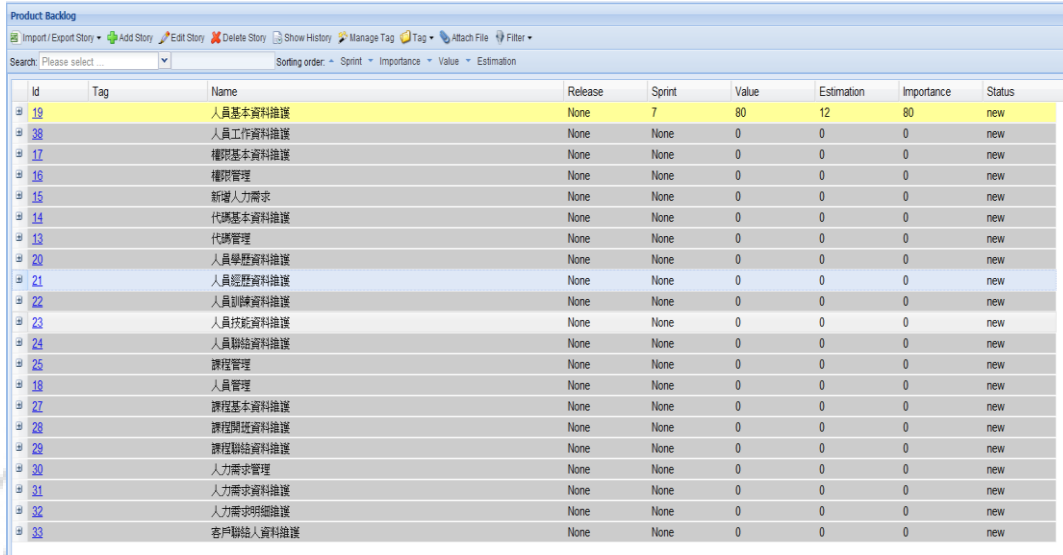
圖 4-40 系統登出功能

資料來源：本研究整理



## 七、精製

產品負責人再經過衝刺所產出的產品檢視後，可以依照可以改善的地方，對產品待辦清單進行修正，或新增某些產品功能。



Id	Tag	Name	Release	Sprint	Value	Estimation	Importance	Status
19		人員基本資料維護	None	7	80	12	80	new
38		人員工作資料維護	None	None	0	0	0	new
17		權限基本資料維護	None	None	0	0	0	new
16		權限管理	None	None	0	0	0	new
15		新增人力需求	None	None	0	0	0	new
14		代課基本資料維護	None	None	0	0	0	new
13		代課管理	None	None	0	0	0	new
20		人員學歷資料維護	None	None	0	0	0	new
21		人員經歷資料維護	None	None	0	0	0	new
22		人員訓練資料維護	None	None	0	0	0	new
23		人員技能資料維護	None	None	0	0	0	new
24		人員聯絡資料維護	None	None	0	0	0	new
25		課程管理	None	None	0	0	0	new
18		人員管理	None	None	0	0	0	new
27		課程基本資料維護	None	None	0	0	0	new
28		課程開班資料維護	None	None	0	0	0	new
29		課程聯絡資料維護	None	None	0	0	0	new
30		人力需求管理	None	None	0	0	0	new
31		人力需求資料維護	None	None	0	0	0	new
32		人力需求明細維護	None	None	0	0	0	new
33		客戶聯絡人資料維護	None	None	0	0	0	new

圖 4-41 精製產品待辦清單

資料來源：本研究整理

## 八、衝刺回顧

每一回合衝刺的結束，會和 Scrum Master 以及產品負責人一起討論本次衝刺的優缺點，以及時間估算的修正，並進行下一回合衝刺改善。



Id	SprintID	Name	Description	Status
Type: Good (3 Items)				
45	7	增刪改查回顧	時限內完成，且順利使用AJAX框架jQuery來達到即時增刪改查功能	new
47	2	AJAX學習使用	由於Struts2有自帶AJAX框架Dojo所以順利達到AJAX功能	new
51	6	登入登出	屬於較容易的功能，所以順利於時限內完成	new
Type: Improvement (4 Items)				
46	1	SSH改版回顧	由於版本差異上的操作過大，導致拖延了一些時限，大部分時間花費於解決函式庫衝突	new
48	3	Struts標籤改成JSTL回顧	由於將所有的Struts2標籤拿掉一項大工程，再加上必須去熟悉另一個AJAX框架jQuery所以時間...	new
49	4	調整檔案結構	為了未來可順利發展此系統，所以花了較多時間於討論何種架構是比較有利，最後也進行結構改...	new
50	5	Maven導入	由於Maven必須與線路有關，但自家線路會不穩，造成浪費許多時間在Maven伺服器架設上	new

圖 4-42 衝刺回顧

資料來源：本研究整理

## 第五章 結論與未來發展方向

本研究透過 Scrum 進行敏捷開發實踐，並輔以 Java 熱門框架技術 Struts、Spring、Hibernate 進行系統開發，提出一套整合性系統發展程序架構，以開發人力資源管理系統雛型為實例說明。

綜合先前章節所述，本研究有以下幾項主要特點：

- 一、技術框架的結合使用：每一種框架都有各自的優缺點，在不同環境所產生的不同需求，整合框架則把框架所擁有之優點結合，但隨之而來的會造成開發複雜度的提升。本研究介紹了 Java 熱門框架 Struts、Spring、Hibernate 以及 AJAX 框架 JQuery 的整合方式，並提出以實作指引文件詳細闡明框架分層原則，降低導入整合框架的複雜度。
- 二、Scrum 敏捷開發流程的具體實現：專案管理方面，本研究實踐了以人為主的 Scrum 敏捷開發流程，以不斷的討論來改進開發程序，並以透過每次的衝刺產生漸增性成果。和過去需要產生大量文件的開發方式相比，Scrum 減少了不需要的文件量，改以直接面對面的溝通解決目前所遇到的問題，每個人以每天移動任務板上的任務來表示任務的執行狀況，而 Scrum Master 則可以透過燃盡圖充分的把握專案工作進度。本研究以開發人力資源管理系統雛型來實踐 Scrum 開發方式。
- 三、高階規劃的統合設計：透過提出系統高階規劃階段，使專案整體透明度提高，並加強專案整體之一致性。本研究在經過系統高階規劃階段所進行的統合設計與開發，對於後續 Scrum 的衝刺的確減少了許多困難點，像是 Scrum 最令人困擾的派工不均，例如資料庫設計、系統分層方式皆在系統高階規劃階段統一完成，程式開發者在衝刺階段只要專注於程式開發，也就解決了派工不均的問題。
- 四、專案管理工具的靈活運用：利用 SVN 對團隊開發程式碼進行版本管理、利用 BugTracker 進行開發中 Bug 議題的追蹤，加上使用 Maven 對專案佈署所需的函式庫進行管理動作，以自動化的方式取代傳統人工的管理，提升開發效率。

關於未來的研究可以分為下列幾點做探討：

### 一、其他專案管理工具的整合

隨著專案規模逐漸增大，系統複雜度也與之俱增，文件檔案的管理、持續整合以及測試自動化，都需要藉由軟體工具來輔助才能有效率的達成，但導入專案管理工具並不是一次將所有的工具一起導入，這樣反而會造成專案團員的額外負擔，故本研究所提出的方法應逐步加入這些自動化工具提升專案管理品質。

### 二、實證方法的驗證

本研究並未對所提出的系統開發程序進行實證方法來驗證，故可對本研究所提出的方法設計一套適用的度量方法或問卷調查，藉以驗證工作時程是否減縮、軟體架構是否具有彈性及擴展性、交付產品品質是否良好等，進行實證研究，以檢視本研究所提出的開發程序效用。

### 三、應用於其他領域

本研究以開發人力資源管理離型系統為例，但所提出的系統開發程序方法架構並未針對特定類型之系統，故可加以延伸於其他各類型應用系統上。

## 參考文獻

1. 方麗玉，2011，國內企業導入人力資源管理資訊系統之研究，國立中央大學人力資源管理研究所碩士論文。
2. 林信良，2008，Spring學習筆記。[Online]. Available: <http://caterpillar.onlyfun.net/Gossip/SpringGossip/SpringABC.html> (Jun. 12, 2012)
3. 吳家豪、余翠瑱、陳建村、鄭有進，2007，JCIS:支援平台相依性建構之Java持續整合系統，第三屆台灣軟體工程研討會論文集。
4. 俞君翰，2002，應用設計樣式於元件式系統開發之研究--以會計總帳系統為例，國立政治大學資訊管理研究所碩士論文。
5. 張弘明，2007，以整合式的教練引導開發以框架為基礎的專案，國立中央大學資訊管理研究所碩士論文。
6. 張嘉琪，2010，一個以Scrum為基礎的軟體工程實務導入方法，國立臺北科技大學資訊工程系碩士班論文。
7. 梁定澎，1997，「資訊管理研究方法總論」，資訊管理學報，資訊管理實證研究方法討論會特刊，第四卷第一期：1-7頁。
8. 葉文崎，2002，元件式系統開發方法研究—以郵務應用為例，國立政治大學資訊管理研究所碩士論文。
9. 薛昊平，2010，人力資源管理系統導入之個案研究—以C公司為例，國立中央大學資訊管理研究所碩士論文。
10. Beck, K., Grenning, J., Martin, R.C., Beedle, M., Highsmith, J., Mellor, S., Van Bennekum, A., Hunt, A., Schwaber, K., Cockburn, A., Jeffries, R., Sutherland, J., Cunningham, W., Kern, J., Thomas, D., Fowler, and M., Marick, B. "Manifesto for Agile Software Development," Agile Alliance, 2001. [Online]. Available: <http://agilemanifesto.org/>(Jun 10, 2012).
11. Boehm, B. W. "A Spiral Model of Software Development and Enhancement," *IEEE*

- Computer* (21:5), 1988, pp. 61-72.
12. ezScrum Official Website, 2011. [Online].  
Available:<http://scrum.tw/index.php/tw/home>(Jun 15, 2012).
  13. Feng, X., and Le, T. “Construction of B2B Electronic Commerce System Based on Apache Struts Framework,” *International Conference on Services Science, Management and Engineering*, 2009, pp 221-224.
  14. Hao, X., and Tang, H. “Struts+Spring+Hibernate Integrated Framework and Its Use in Log Accounting and Analyzing System,” *International Conference on Multimedia Information Networking and Security*, 2010, pp 936-939.
  15. Hibernate Official Website, 2012. [Online]. Available:<http://www.hibernate.org/>(Jun 18, 2012).
  16. Kniberg, H. *Scrum and XP from the Trenches: Enterprise Software Development* Lulu.com, 2007.
  17. Kruchten, P. *The Rational Unified Process : An Introduction* (3rd Edition), Boston: Addison-Wesley, 2004.
  18. Liu, D. “Design and Implementation of High-Quality Course Scoring System Based on Struts and Spring and Hibernate Architecture,” *International Conference of Information Technology, Computer Engineering and Management Sciences*, 2011, pp 46-48.
  19. Liu, F., Guo, H., and Fu, B. “The Research of Web Application Framework Based on SSH,” *International Seminar on Business and Information Management*, 2008, pp 169-172.
  20. Martin, R. C. *Agile Software Development: Principles, Patterns, and Practices* Prentice, Hall PTR, Upper Saddle River, NJ, USA, 2003.
  21. Mills, H. *Top-down programming in large systems* Prentice-Hall ,USA,1971.
  22. Morton, S., “*The State of the Art of Research, The Information System Research*

- Challenge*,” edited by W. McFarlan, Harvard Business School Press, Boston, 1984, pp. 13-41.
23. Object Management Group, 2001. [Online]. Available:<http://www.omg.org/>(Jun 6, 2012).
  24. Rising, L., and Janoff, N. S. “The Scrum Software Development Process for Small Teams,” *IEEE Softw.* (17:4), July 2000, pp 26-32.
  25. Royce, W. W. “Managing the development of large software systems”, *Proc. Westcon*, IEEE CS Press, 1970, pp. 328-339.
  26. Scott-Morton, M.S. “The State of the Art of Research,” *The Information Systems Research*, Poston: Harvard Business School Press, 1984, pp.13-41.
  27. Shan, C., and Yuntao, J. “The Discussion on Improvement of the J2EE Framework Based on SSH,” *International Conference on Future Computer Science and Education*, 2011, pp 518-521.
  28. Spring Official Website, 2012. [Online]. Available:<http://www.springsource.org/>(Jun 8, 2012).
  29. Struts Official Website, 2012. [Online]. Available:<http://struts.apache.org/2.0.6/docs/architecture.html>(Jun 19, 2012).
  30. Sutherland, J., Schwaber, K. “The Scrum Papers: Nuts, Bolts, and Origins of an Agile Process.,” Draft, 2007. [Online]. Available:<http://scrumtraininginstitute.com/>(Jun 21, 2012).
  31. The Spring Framework - Reference Documentation, 2012. [Online]. Available:<http://static.springsource.org/spring/docs/2.0.x/reference/index.html>(Jun 20, 2012).
  32. Williams, L., Brown, G., Meltzer, A., and Nagappan, N. “Scrum + Engineering Practices: Experiences of Three Microsoft Teams,” *International Symposium on Empirical*

*Software Engineering and Measurement*, 2011, pp 463-471.

33. World Wide Web Technology Surveys, 2012. [Online].

Available:[http://w3techs.com/technologies/cross/javascript\\_library/ranking](http://w3techs.com/technologies/cross/javascript_library/ranking)(Jun 5, 2012).

34. Yongchang, R., Xing, T., Xing, Z., and Zheng, J. “Application Research for Integrated SSH Combination Framework to Achieve MVC Mode,” *International Conference on Computational and Information Sciences*, 2011, pp 499-502.

35. ZeroTurnaround, 2011. [Online].

Available:<http://zeroturnaround.com/java-ee-productivity-report-2011/>(Jun 2, 2012).

